DOCUMENT RESUME

ED 243 455 IR 011 069

TITLE Computers in the New Curriculum. Proceedings of the Annual Conference of the Texas Computer Education Association (4th, Austin, Texas, February 29-March 3, 1984).

PUB DATE 84

NOTE 249p.

PUB TYPE Collected Works - Conference Proceedings (021) -- Viewpoints (120) -- Reports - Descriptive (141)

EDRS PRICE MF01/PC10 Plus Postage.

DESCRIPTORS *Computer Assisted Instruction; *Computer Literacy; *Computer Managed Instruction; *Computers; *Computer Software; Educational Administration; Elementary Secondary Education; Program Descriptions; Program Implementation; Programing; Staff Development; Teacher Education

IDENTIFIERS *Computer Uses in Education

ABSTRACT

This 63-paper collection represents a variety of interests and areas of expertise related to technology and its impact on the educational process at all levels. Topics include automated instructional management, computer literacy, software evaluation, beginning a computer program, finding software, networking, programming, and the computer and educational administrators. Uses of computers are described for staff development and for teaching study skills, writing, social studies, and mathematics. Additional papers describe projects involving district plans for computer literacy, a teacher technologist training program, automating a library management system, creating slides with graphic software, microcomputer adaptive diagnosis for mathematics, and the application of artificial intelligence in education. PILOT and LOGO are among the programming languages discussed. Most papers include abstracts and references. (LMM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

ED243455

COMPUTERS IN THE NEW CURRICULUM

Conference Proceedings

the Fourth Annual TCEA State Conference

Austin Hyatt-Regency Austin, Texas

February 29 - March 3, 1984

IR011069

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Philip F. Gehring

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

TEXAS COMPUTER EDUCATION ASSOCIATION

Board of Directors

1983-1984

PRESIDENT
Sandy Pratscher
Texas Education Agency

VICE-PRESIDENT
Vicki S. Smith
Region IV Education Service Center, Houston

PAST PRESIDENT
Lyndal R. Hutcherson
Carrollton - Farmers Branch ISD

SECRETARY
Karen L. Jones
Round Rock ISD

TREASURER
Diana Radspinner
Dallas ISD

NEWSLETTER EDITOR
Kathy Fite
Southwest Texas State University

AREA I DIRECTORS
Byron Craig
Abilene ISD

Gene E. Rooze
Texas Tech University

AREA II DIRECTORS
Keith Mitchell
Texas Education Agency

Ron King
Baylor University

AREA III DIRECTORS
E. Gillette Irby
Corpus Christi ISD

Keith Imon
Region XX Education Service Center, San Antonio

AREA IV DIRECTORS
Dede Heidt
Fort Bend ISD

Jennabeth Bogard
Cypress-Fairbanks ISD

AREA V DIRECTORS
Bill Adkins
Plano ISD

Judith A. Leyrer
Carrollton - Farmers Branch ISD

FOREWARD

The Texas Computer Education Association is a young but rapidly growing organization in the state. From its inception in 1980, it has grown to a membership of about one thousand.

TCEA is dedicated to:

(1)encouraging an active interest in computer education on levels K-12,
(2)providing an opportunity to study and keep abreast of any new trends in education,
(3)improving teacher training programs in the area of computer education,
(4)promoting professional cooperation and communication among teachers and administrators toward the realization of sound educational achievements, and
(5)providing a liaison with other organizations involved in the use of the computer as an instructional tool.

The Texas Computer Education is an organizational member of the International Council for Computers in Education (ICCE), and during the 1983-1984 school year, it expanded its role in Texas schools. Awards are being given at the annual conference for recognition of winners of area and state programming competition for high school students, winners of the state computer literacy contest for junior high students, and teacher-developed instructional materials and software. The purpose of all competition organized through TCEA is to recognize those students and teachers who are doing outstanding work in the field of computing. TCEA considers it important that teachers be recognized for their exemplary work and that students pursuing academic coursework be congratulated and recognized for their efforts.

TCEA is proud of its efforts on the Joint Texas Association for Educational Data Systems (TAEDS)/Texas Computer Education Association Statewide Committee to Promote Computer Literacy, which made significant strides in causing computer literacy to be recognized as an essential part of education for all Texas students. The committee published and disseminated a position paper which was recognized by the National Science Foundation; emdorsed by a host of individuals and groups, including Governor Mark White, the Texas Congress of Parents and Teachers, and over twenty professional Texas education societies; and supported by the legislative Subcommittee on Changing Technology in Instruction.

TCEA communicates to its members through a newsletter as well as on an electronic bulletin board service, which includes reduced subscription fees for members.

Special interest group contacts have been identified to better accomplish delivery of information, research, and identification of needs in computing for specific content areas.

This proceedings book represents a variety of interests as well as areas of expertise related to technology and its impact on the educational process. I would like to express my appreciation to those presenters who submitted papers for this book. I would especially like to thank all of those who worked so tirelessly to ensure the success of the TCEA Fourth Annual Conference.

The Texas Computer Education Association is proud of its membership as well as of its sensitivity to the needs of educators and students across the state. We're growing ---we're caring---we're accomplishing.

Vicki S. Smith
Conference Program Chair

PAPERS BY AUTHORS

ACKELSON, JOHN
ROGERS, DR. TED
Automated Instructional Management AIM/3000..... 1

ADKINS, BILL
Now That I Have 'Em, What Do I Do With 'Em?..... 5

ARCH, JOHN, PH.D
Why Computer Literacy?..... 7

BAIRD, BILL
Evolution of a Microcomputer Laboratory for
Teacher Training and Staff Development..... 9

BANE, ROBERT K., PH.D
HRESKO, WAYNE P., PH.D
Software: Copyrighting, Protecting, Pirating..... 15

BANE, DR. ROBERT K.
SCHNEIDER, JUDITH
SPENCE, M. JANET
POPE, CAROLYN
RIGG, LYNNE
Teachers and Computers: What Knowledge Is of Most Worth?..... 19

BASEL, JOSEPHINE
Beginning a Computer Program..... 24

BASEL, JOSEPHINE R.
Networking With the TRS-80..... 27

BASEL, JOSEPHINE R.
Staff Development..... 31

BERRY, JULIA
KRAMER, PAT
Input: Planning and Staff Development
Output: A Successful K-12 Computer Curriculum..... 34

BRAZIER, PEARL W.
Interchanging Data and Programs Between
Microcomputers and Mainframes..... 36

BROWN, DR. DOUGLAS L.
Teaching PASCAL: Who, What, When, Where, How...and Why?..... 39

BUCHAN, HELEN TEMPLE
A Complete Computer Literacy Course Presented Via CMIVI..... 43

CHARLES, HARRY
HAVEN, DR. SANDRA
Implementation of a Computer Literacy Unit
For Eighth Grade Middle School Students..... 48

CHILTON, BRAD, ED.D
Guidelines for Success: The Training of Future and Practicing Teachers in Computer Usage..... 51

CHILTON, BRAD, ED.D
Microcomputer Usage and Utilization in North Central Texas..... 54

CHILTON, BRAD, ED.D
Purchasing a Microcomputer?--Be Careful--Don't Get an Edsel..... 56

CHILTON, BRAD, ED.D
Readin', 'Ritin', and 'Rithmetic Move Over For a New Basic..... 59

CHILTON, BRAD, ED.D
A Time of Change: Computers Challenge Today's Educational Administrators..... 61

FROESE, K. LYLE, ED.D
Microcomputer Literacy for the Educational Administrator: Applications for Microcomputer Systems for Noninstructional Tasks..... 63

GADZELLA, BERNADETTE M., PH.D.
Computer-Assisted Instruction (CAI) Study Skills With High School Students..... 67

GEHRING, DR. PHILIP
The Fifth Generation and Expert Systems..... 72

GERSHNER, VERA T., PH.D.
LAWTON, JOHNNY
Teachers: The Key to Computers in the Curriculum..... 80

GOLDBERG, DEE
MATTHEWS, FRANK F.
Comparing LOGOs..... 85

HALVERSON, RANETTE H.
Nassi-Shneiderman Diagrams-An Alternative to Classical Flowcharting..... 90

HARVEY, WILLIAM J., PH.D
Random Selection Algorithms for Computer-Based Instruction..... 95

HAWKINS, M.
NIGLIAZZO, M.
STONE, V.
DEMAREE, W.
SPHAR, C.
TOMLINSON, H.
The English Learning Center System..... 101

HINES, SANDRA N.
The Thinking of Five-Year-Old LOGO Programmers..... 104

HOOT, JAMES L., PH.D
Improving Writing in the Primary Grades Through Computer Technology..... 107

HUBER, JOHN
Computer Mathematics..... 110

HUBER, JOHN
Interfacing an Optical Reader With a Microcomputer.... 112

HUSTING, BARBARA
WEBB, SANDRA
Biography of an Elementary Library Learning Center-
Computer Lab.... 114

IRBY, E. GILLETTE
A District Plan for Computer Literacy
Design and Implementation.... 120

JOSLIN, ELLEN
Make It Move!.... 124

KIMBROUGH, NANCY
A Teacher Technologist Training Program.... 128

KNEZEK, DON
Sports Related Microcomputer Use With Emphasis on
Football Scouting and Analysis.... 132

LEE, ROBERT, JR.
Alternative High School Use of Microcomputers.... 136

LOGUE, GEORGE A.
Computer Literacy: An Integrated Approach.... 141

LUPER-FOY, SUSANN
Automating a Library Management System.... 144

MILLER, JEFF
How We Did It: San Angelo's 1st Year-6th Grade
Computer Awareness Program.... 150

MUNCRIEF, DR. MARTHA
Securing Teacher Acceptance of Technology.... 152

NIX, LINDA
Teaching Basic Math Skills Through Computer Concepts.... 156

NORTHRUP, DR. TERRY
ROOZE, DR. GENE E.
Sniffing Out Good Software Programs: Using
the NOSES Program.... 159

PIASECKI, FRANK, PH.D.
Learning How to Use Delta Drawing.... 163

PIPER, KAREN L.
Teaching the Teachers: The Key to Instructional
Word Processing.... 166

POLLARD, GERALD
How to Create 35mm Color Slides With Graphic Software
and Your Home Computer.... 170

POTTER, RITA
The History of the Computer: A Model for Infused
Ojectives in Social Studies.... 172

PROCTER, DANNAH
Student Use of Data Management Program: New Tool for Extended Study Skills..... 175

RAMPY, DR. LEAH
SADOWSKI, DR. BARBARA
SIGNER, DR. BARBARA
Planning for Computer-Fluent Teachers..... 178

ROOZE, GENE E.
NORTHRUP, TERRY
Social Studies, the Computer and H.B. 246..... 183

SADOWSKI, BARBARA R.
Designing an Integrated K-12 Computer Curriculum..... 188

SCHAFFER, ROBERT E., PH.D.
A Computer-Based Electrophysiological Laboratory for High School and College..... 193

SCLAFANI, SUSAN K.
Planning Your School Computer Program..... 199

SEMMES, DR. PAT
DEWOLFE, CAPTAIN JACKIE
RODE, PAUL
CALDWELL, JILL
DILLINGHAM, DR. KAY
Implementation of the Little LOGO Curriculum in the Fourth Grade..... 203

SEMMES, PAT
The EGG Computer: A Simple Model..... 205

SIGNER, DR. BARBARA
MATH DOCTOR, M.D. Microcomputer Adaptive Diagnosis..... 209

SIGNER, DR. BARBARA
Computer Literacy for Teachers..... 213

SMITH, RICHARD ALAN, PH.D.
Controlling the Computer Demonstration Process..... 216

SMITH, VICKI S.
Software Evaluation: Support Through the Texas Education Computer Cooperative (TECC) Project..... 219

SWIGGER, DR. KATHLEEN M.
The AI in CAI:
The Application of Artificial Intelligence in Education..... 222

SWIGGER, DR. KATHLEEN M.
Teaching Beginning Students How to Program..... 227

WENDT, JANICE C,
MORROW, JAMES R., JR.
Administrative Applications of Microcomputer Software for Physical Education, Sports and Athletics..... 232

WILSON, JEAN W., ED.D
Is There a Place for VISICALC in the Elementary School Classroom?..... 238

AUTOMATED INSTRUCTIONAL MANAGEMENT
AIM/3000

John Ackelson
Dr. Ted Rogers

School District No. 12, Adams County
Northglenn, Colorado

ABSTRACT

The AIM/3000 system was designed in order to facilitate the management of instruction at the classroom level in a mastery learning environment and to allow aggregation of data for use at various levels within the organizational structure. The system was further designed to interface with other major systems within the overall management information system. The system components include: (1) instructional objectives data bank; (2) instructional resource system; (3) test item bank; (4) student diagnostic capabilities; (5) student progress reporting system; and (6) aggregate of data at defined levels. The perceived benefits of the system for instructors include: (1) reduction in time for record keeping activities; (2) reduction in time in locating resources needed for instruction; (3) generation of exams; (4) scoring of exams; and (5) providing data for managing classrooms in terms of levels of achievement by students. The system is being implemented in six beta sites which includes two elementaries, two junior highs and two high schools. Data gathered from the beta sites will be evaluated in terms of instructor's time necessary to operate the system, utilization of the system and influence on student achievement.

School District No. 12, Adams County began the development of instructional objectives and methods of measuring the achievement of students in relationship to these objectives in 1973. Since then, the district has developed and refined instructional objectives in the curricular areas of mathematics, reading, language arts, social studies, science, foreign language, music, art, health and physical education.

The measurement techniques associated with each objective have been developed and examined for validity and reliability. As the objectives are reviewed and further refined, so are the measurement methodologies.

As instructors taught to the instructional objectives, several problems emerged in terms of management or instruction. Some of these problems included:

1. Record-keeping of student achievement of the instructional objectives.
2. Development of reliable and valid exams for formative evaluation purposes.
3. Scoring of examinations.
4. Obtaining diagnostic information concerning student achievement in regard to process (thinking skills) and content (subject matter).
5. Correlating instructional objectives from one curricular area to the other, so redundant instruction did not occur.
6. Reporting to parents concerning mastery of objectives for students.
7. Locating resource material for teaching of the instructional objectives and a description of the resource material, in terms of its relation, to the objectives.

The solution in regard to these problems, was an automated system. However, for a system to be able to address the described problems, there were a number of parameters that had to be considered. These parameters included:

1. The system had to be able to accommodate the data requirements of 1,000+ instructors and 19,000+ students in various curricular areas.

2. The system had to have the capacity of maintaining records in the ten curricular areas for the student's entire stay in the district, be it for 1 month or thirteen years, and be capable of being accessed by instructors who need to examine the data.
3. The system had to have the capability of generating various aggregations of data, based on the organizational structure, and its need for data. Therefore, the system had to aggregate data at the (a) classroom level, (b) grade within a school level, and (c) grade within the district level.
4. The system had to be very user friendly.
5. The system had to be integrated into the MIS and be interfaced with the student information system, summative testing system, library system, attendance system, grading system, and financial system.

As the district began to address the problems and the above defined parameters, a number of existing systems were examined by a team of instructors, administrators and system analysts. This process began in 1980 and ended in October of 1982.

The major CAI/CMI systems were examined, in detail, as well as a number of micro-systems. The major conclusions of this 2 1/2 year study were:

1. The examined systems were not compatible with the district MIS system unless extensive and complex modifications were made.
2. The examined systems were designed for training environments and not a public education environment.
3. There were no provisions for mass data input.
4. No provisions were made for aggregation of data beyond the classroom.
5. The hardware necessary to drive the systems was very expensive.
6. The examined systems could not address the district library-media resources and the necessary information needed for each.

With these conclusions, it was decided to develop a CMI system for use in the public education environment. Financial support for the development of the system came not only from district funds, but from Hewlett-Packard and, to a lesser degree, National Computer Systems.

The system was designed with input from over 130 people, including classroom instructors, school administrators, curriculum specialists, media specialists, district administrators, school board members, system analysts and hardware and software vendors.

The system is currently being tested at six beta sites consisting of two elementaries, two junior high schools and two high schools. Data from the beta sites will be available in April.

Data analysis will focus on three elements. These include: (1) instructor time necessary to operate the system, (2) utilization of the system by the instructors, and (3) student achievement.

OVERVIEW OF AIM/3000

The use and operation of AIM/3000 is centered around a number of system components. The primary focus of AIM/3000 is

directed at instructors and their management of instruction.

The major components of the CMI System include:

1. Instructional Objectives Data Bank,
2. Instructional Resources Data Bank,
3. Test Items Data Bank,
4. Student Diagnostic Capabilities,
5. Student Data Bank/ Student Progress Reporting, and
6. Aggregation of Data at Various Levels.

INSTRUCTIONAL OBJECTIVES

For each instructional objective in the system the following information is displayed:

1. A brief description of the objective,
2. A complete description of the objective,
3. A taxonomical classification for the objectives,
4. Skill level,
5. Difficulty level,
6. Mastery criteria,
7. Correlated instructional objectives in the same or other curricular areas, and
8. Number of items in the item bank for the objective (this element is system generated).

RESOURCE DATA BANK

The resource data bank allows instructors to locate instructional material, based upon a particular instructional objective.

When an instructor wishes to locate resources, the CMI System prompts the instructor for the short code of the desired instructional objective. The objective description is retrieved and displayed, along with the available resources and their learning modes and locations within the school district.

TEST ITEM DATA BANK

Instructors will utilize the TEST ITEM DATA BANK to generate tests to measure the mastery level of their students, in relation to specific instructional objectives. The AIM/3000 System will prompt an instructor to identify the instructional objectives to be tested, the number of test items to be included for each objective, the types of questions to be included (TRUE/FALSE, MULTIPLE CHOICE, etc.) and the range of the difficulty of the items. To aid the instructor in assessing the difficulty level of the items, a continuous record is kept that displays the number of responses for each foil for each item. The CMI System selects items from the TEST ITEM DATA BANK that meet the instructor's criteria, and transmits them to the printer at the instructor's school, along with the appropriate key.

In addition to the system generating the exam for the instructors, three other methods for test definition are available. The first method is a manual selection of each item as desired on the exam by the instructor. The instructor simply enters into the item bank and selects the exact item they desire for the exam. The second method is that an instructor can develop a teacher-made exam and then define the exam in terms of the objectives tested and the relationship of the items to the objectives within the system. The system will then score the exam, update the student records as appropriate and generate reports from the testing, as if it was a system-generated exam. The third alternative is to use a previously defined exam in the test library of the system.

Students respond to the question on Optical Mark Sense scan sheets, which are transmitted back to the central computer for updating

of the CMI Data Base. In addition to updating the CMI Data Base for each student, diagnostic and item analysis reports are pro duced and transmitted back to the instructor's school for on-site printing. The student diagnostic report is capable of providing process, as well as content data, back to the instructor.

Three Test Analysis Reports are produced with the scoring of each CMI test:

1. Frequency Distribution Report,
2. Student Mastery Item Analysis Report, and
3. Student Diagnostic Report.

STUDENT DATA BANK

The AIM/3000 maintains basic information about each student in the district. This includes for example, student name, grade level, sex, school I.D., enrollment date, home room number, and the various courses that a student is presently enrolled. In addition, the AIM/3000 System maintains instructional mastery information for all students. The system maintains for the current year, the instructional objectives that each student has mastered, including the date of this mastery. The system also maintains mastery history for each student, identifying by sub-curriculum areas (e.g., Reading-Grade 3), those instructional objectives that a student has mastered during prior years.

STUDENT PROGRESS REPORTING

The mode of data input from the various schools for updating student mastery of objectives varies. In most cases, data entry occurs through an Optical Mark Reader located at the school site. In some cases, the data entry is on-line through a CRT.

The type of mastery data transmitted through a school's Optical Mark Reader generally is student test data. Teacher observation data is also sent to the central computer; some objectives are not measurable by pencil and paper; however, an instructor may be able to observe behavior to determine mastery. These observations still come into the system as mastery of objectives and update student mastery; however, no scoring is necessary as is the case with student test data.

In addition to the on-line displays for student progress, four Student Progress Reports are available through the CMI System:

1. Student Mastery Report,
2. Class Mastery Report,
3. School and District Mastery Report by Grade, and
4. Percentage of Students Mastery Report, Total District.

CONCLUSION

The AIM/3000 System was designed to meet the needs of instructors who are engaged in teaching to defined instructional objectives and are involved with on-going assessment of student achievement concerning mastery of the instructional objectives. It was also designed as part of an overall education MIS, which would allow decision makers, at various levels within the organization, to access pertinent and timely data.

The system should reduce significantly the instructor's time needed to perform clerical tasks, in relationship to the district's mastery learning system. This will be determined by April, as beta site testing is completed.

In brief, the perceived benefits to the instructors include:

1. Reduction of time in locating resources needed for instruction,
2. Generation of exams,
3. Scoring of exams, and
4. Providing data for managing their classroom in terms of achievement.

NOW THAT I HAVE 'EM, WHAT DO I DO WITH 'EM?

Bill Adkins
Plano Independent School District
1517 Avenue H
Plano, Texas 75074

During the 1983-84 school year, every fourth, fifth, sixth, and eighth grader is working with the computer as part of the regular instructional program in the Plano I.S.D. Fifth and eighth graders are using CAI programs in mathematics; fourth and sixth graders are using them in language arts. Details are presented in this session.

Although there are many uses for the computer, the Plano I.S.D. has targeted computer assisted instruction. The CAI program is a highly centralized program which is distributed to individual schools for implementation. Both hardware and software is selected by a central study committee composed of teachers, principals, and central office staff.

Each year, a specific area is targeted (example: fourth grade language arts). After selecting the subject, the appropriate coordinator meets with teachers from the targeted area to outline goals, select materials, write guidelines, and train staff. Hardware and software is distributed to each campus based on the population of the school.

During the 1983-84 school year, the Plano I.S.D. targeted fourth grade language arts, fifth grade mathematics, sixth grade language arts, and eighth grade mathematics. All of these programs required that each student receive thirty minutes per week via the computer. This usually is accomplished through two fifteen minute sessions.

Criteria for selection of materials for these areas included the following:

- -match between the software and Plano I.S.D. instructional objective
- -match between the software and the instructional level
- -suitability of content
- -interest level

The software was selected by appropriate subject matter coordinators and representative teams of teachers.

In the fourth grade language arts program, students are rotated through programs in vocabulary development, spelling, language mechanics, language usage, study skills, and comprehension. Ten software packages from six vendors are used to present the material to students. The computer programs are used in conjunction with material which has been introduced by the teacher in the classroom.

The fifth grade program targets mathematics. Since Plano I.S.D. has adopted a continuous progress program of mathematics, the software chosen had to cross several grade levels. Two

packages from two vendors are the basics for the mathematics program. The emphasis is on drill and practice, with some tutorial work being done. Students are assigned lessons based on identified weaknesses. The strength of the program is the immediate response the student receives, the help provided by the computer, and the inherent motivation of using the computer.

The sixth grade program introduces students to basic keyboarding skills. Following that instruction, students are taught to use a simple word processing package as a tool for developing composition skills. Emphasis is placed on sentence combining techniques and developing improved finished products. Students are more willing to edit when they don't have to rewrite material. In addition, software programs to supplement various areas of the language arts are provided for student use.

The eighth grade program is very similar to the fifth grade program. The middle schools group students according to ability. Computer time is allocated to students based on identified needs.

Because of the numbers of students compared to the number of computers, the eighth grade used the computers the first semester; the sixth grade uses them the second semester.

Each of the programs is designed to meet the following goals:

- to introduce students to the computer.
- to develop confidence in using the computer as an educational tool.
- to develop independent work habits.
- to provide practice in following instructions in a structured way.
- to provide practice in thinking skills.
- to individualize assignments.
- to meet specific objectives in the content area.
- to provide variety in the instructional process.

Evaluation of the program has been done in several ways. In 1981-82 students were compared through experimental-control groups. In 1982-83, a visiting team of outside consultants spent a day evaluating the program. For the 1983-84 school year, several studies are in progress to evaluate the program.

WHY COMPUTER LITERACY?

JOHN ARCH, Ph.D.
HOUSTON INDEPENDENT SCHOOL DISTRICT
DEPARTMENT OF TECHNOLOGY
5300 SAN FELIPE
HOUSTON, TEXAS 77056

ABSTRACT

Many school districts across Texas and the nation are scurrying to implement a computer literacy curriculum. This session seeks to examine the reasons for developing a pre-college computer literacy curriculum.

The intended audience for this session will be curriculum developers at all levels, administrators, classroom teachers, and other interested educators. No computer expertise will be assumed.

Many school districts across Texas and the nation are spending increasingly large sums of money on something called computer literacy. In this presentation, I will examine some of the reasons for this phenomenon.

First, some examples. Recently, the governor of Tennessee, Lamar Alexander, called for increased expenditures on computer science education, with the eventual goal of placing at least one microcomputer and a teacher trained to use it in every school in Tennessee. By the 1985-86 school year, every seventh and eighth grader in the state will have to take computer literacy. This from a state not known for leadership in the area of technological education.

Other states and localities with ambitious programs include Florida, Boston, Cupertino (California), and Montgomery County, Maryland.

In 1982, the Houston Independent School District created the nation's first Department of Technology. With an annual budget of over several million dollars and a staff of over 40 professionals, the DOT has the dual tasks of overseeing the delivery of computer literacy to Houston's 190,000 students and training the District's 14,000 teachers. The local school board has mandated that by 1986, every graduate of an HISD high school must be certified as computer literate.

The time has come to ask why these expenditures are being made. Why should we, as educators, be so concerned with computer literacy? There are several important reasons often mentioned in the literature.

We are under considerable pressure from parents, politicians, even television commercials, to get our act together and do a better job educating students. Computers are seen as an important new tool for teachers in this struggle. The recently published A NATION AT RISK details the scope of the problem. Lower test scores, lower standards for graduation--in general, a perception that schools are not as rigorous or good as they once were. When compared with students from other countries, American children come out at or near the bottom in most academic areas. Computers are seen as a way to help alleviate these poor performances. There is a growing body of evidence that CAI can help certain students learn certain things

faster than they can learn them via traditional methods. Concerned parents and educators naturally want to explore the possibilities of using computer technology to augment the teacher's power.

A second reason given for stressing computer literacy is the change in employment opportunities being brought about by the computer. It has been estimated that by the year 2000, ninety percent of all jobs will be computer-related. It is natural for parents and educators to be concerned with these changes and to want to prepare today's students for future job markets.

Another commonly stated reason for teaching computer literacy is to help students increase their problem-solving abilities. I define problem-solving skills in a broad sense. The definition I use is close to what Robert Taylor terms using the computer as a tool. Using this definition, word processing, programming, spreadsheets, and so on are seen as new ways for students to solve problems. Productive citizens in the future will have to gain competence with the new problem-solving tools made possible by the computer.

There are other reasons given for studying computer literacy, but the three mentioned here are among the most widely used. To sum up so far: among the most common reasons for teaching computer literacy are:

(1) to improve student achievement
(2) to prepare students for tomorrow's jobs, and
(3) to improve students' problem-solving abilities.

Each of these reasons is a worthy goal in and of itself. Who could possibly be against any of them? But, there are others. One reason for teaching computer literacy that is often overlooked is one that we are trying to stress at HISD. This is the profound change that computers are making and will continue to make upon the way we perceive the world. Understanding computers is important because computers are important to our society. Would there be an uproar about making all students computer literate if the computer were as limited in its effect upon society as a toaster? Do you hear anyone demanding that all students be made toaster literate? Of course not. Computers are not just another gadget. They are far more important to our society than filmstrip projectors and most other examples of educational technology. The only piece of technology that has had such a global effect upon our society is television and, while the impact of educational television has been disappointing, the impact of TV upon the minds and attitudes of our students has been immense.

So it will be with computers. Think of the topics that confront us that are deeply intertwined with computers. The space program, most scientific research, the competition between us and Japan, the arms race, the impact of industrial robots upon employment, and so on. A computer literacy course should deal with these issues. It should be used as a vehicle to get students to think about them. Computer literacy is much more than programming, base 2 arithmetic, and Charles Babbage.

Computers have changed and will continue to change the way we perceive the world. Each of our students will have to learn to cope with them. It is the job of computer educators to constantly re-evaluate the effects computers are making on our society and then to ask how the curriculum should be changed to reflect these effects.

Evolution of a Microcomputer Laboratory for Teacher Training and Staff Development

Bill Baird
Learning Resources Center, EDB 536
University of Texas at Austin 78712

ABSTRACT

The Learning Resources Center of the College of Education, University of Texas at Austin, includes as part of its mission the training of teachers in effective use of media. For the past four years the media center has included a microcomputer laboratory. Within the past year this facility has grown quite rapidly in both equipment and software holdings. A variety of courses use the lab facilities for preservice and graduate education. Open orientations for faculty and students are offered on a regular basis. Evaluations of these sessions and a survey of lab usage indicated types of needs and degree of satisfaction by clients.

PREFACE

The explosive growth of microcomputer applications at the pre-college level since the mid-70's has caught most teacher training institutions by surprise. Reduced costs of small computers and the availability of quality courseware have resulted in unprecedented demand for training teachers and administrators in ways to apply the new technology. David Moursund notes that better hardware and courseware can be mass produced and mass distributed, but "without knowledgeable and supportive teachers the results will be disappointing and often a failure." (Moursund, 1981)

The Learning Resource Center of the College of Education at the University of Texas contains an educational library and a media center. Its mission includes providing print materials, media equipment and training for staff and students of the College. It is open Monday - Thursday from 9:00 am until 9:00 pm, on Friday from 9:00 am - 5:00 pm, and Saturday from 9:00 am until noon. The media lab is staffed by lab assistants who take reservations for equipment by phone, provide assistance, and sell supplies. One component of the media center since 1980 has been a microcomputer laboratory. This area has a separate staff for technical support. Basic administration is carried out by media lab assistants, who provide orientations in addition to routine supervision.

HISTORY

Prior to December 1982, the Microcomputer Lab contained five TRS-80 Model I microcomputers with 16K and cassette tape drives; two 48K Apple II+ machines, with dual disk drives; and an IDS-460 dot-matrix printer which could be connected to either a TRS-80 or

one of the Apple machines. Software consisted mostly of material from the Minnesota Educational Computing Consortium (MECC) project and cassette programs for the TRS-80 machines. One teaching assistant was hired in the summer of 1982. This graduate student supervised for 20 of the 60 hours that the lab was open and provided four orientations to groups each semester.

The primary uses of the lab were support of one programming class in BASIC at the graduate level and brief orientations during the instructional media classes. There was limited use of microcomputers in other departments. An Instructional Systems Laboratory on the same floor provided terminals for access to mainframe computers elsewhere on campus.

CURRENT STATUS

Recent arrivals of new equipment and software have resulted in dramatic expansion of the microcomputer lab facilities. It has been necessary to expand into a converted television studio to accommodate thirty-six new Apple IIe machines. Except for the eighteen stations with color monitors, these machines come with 80-column screen displays. All stations have 64K of memory and two disk drives. The new lab annex has been equipped with a mounted 35-mm slide projector and overhead projector to facilitate presentation of classroom and orientation sessions. Two new daisy-wheel printers when they arrive will provide clients with three letter-quality printers for word processing. Four additional dot-matrix printers are on order.

Software holdings have also grown during the past year. As money becomes available, new educational courseware which has received favorable reviews is ordered. The old courseware file boxes in which software was cataloged by title and subject have recently been replaced by a computerized file management system. All software is cataloged using PFS - a file management program which allows searching all holdings by any key field of each holding record. There are over 100 software "packages", some consisting of several disks and/or documentation manuals. A software cataloging system was developed by Jay Fermaglich - Media Lab Supervisor. Software is catalogued under the following headings:

1) authoring systems
2) business
3) computer-assisted instruction
4) computer literacy
5) data management
6) demonstration
7) early childhood
8) English
9) games
10) graphics
11) language
12) math
13) music
14) programming

15) reference
16) science
17) simulation
18) social studies
19) special education
20) utility
21) vocational education
22) word processing

Each record in the catalog contains the following fields:

1) NAME [title of package]
2) PUBLISHER [source]
3) SUBJECT [one of the above 22]
4) CROSS-REFERENCE [usually several]
5) DESCRIPTION [usually 2-3 paragraphs]
6) DISKETTE NAME(S) and SLEEVE NUMBER(S)
7) DOCUMENTATION NAME(S) and NUMBER(S)

Large three-ring notebooks contain the above record for each of the holdings. One notebook contains holding records alphabetically by title; two more are listed by subject and contain several cross references for most items. These notebooks are available at the checkout desk, with backup copies in a staff room. Each courseware item can be checked out from a locked room behind the media center desk for use within the lab. None may be taken out of the media area. No duplication of copyrighted material is allowed. However, backup copies of all holdings are kept locked in a separate room.

In the fall semester of 1983 over 600 students in at least seven different courses made use of the lab facilities. These courses included computer literacy, BASIC programming, special applications, CAI design, advanced media and survey media (required). All sections of these classes were filled, and many students were turned away. More courses are planned, including a Pascal programming class and advanced BASIC to be offered in the spring semester of 1984.

The Texas Board of Education is considering a mandate that all teachers certified in 1984 must complete at least one course in the use of computers. Clearly, the demand for teacher training in computer applications is growing! The most likely training center for teachers will be colleges of education, since computer science departments are already over-subscribed and under-staffed. So the question becomes how to provide the best access and training with available staff and equipment?

MANAGEMENT AND ACCESS

The value of peer interaction among teachers learning to use computers has been noted elsewhere. (Wells and Bitter, 1982) Thus, the physical arrangement of equipment, provision of group orientations and supervision of the lab area are all designed to encourage cross-pollination of ideas and mutual support among users of the facilities. This philosophy has the added advantage of maximizing efficient uses of limited resources.

Equipment is available in carrels, which are 36" wide by 24" deep by 19" high, and provided with 110-volt grounded outlets. Table surface is 29" above the floor. The main lab area is in a 24 by 39-foot room. This commons area has thirteen carrels plus three printer stations. The adjoining studio lab contains 30 of these carrels and has dimensions of 18 by 28 feet. Teaching assistant offices open into the commons area. Plans are to connect each printer to four switch-selectable work stations.

There are currently five part-time teaching assistants for the microcomputer lab. These graduate students work from 8 to 20 hours per week. Their duties include assisting clients with any problems involving disks, printers, disk drives, in addition to providing regular orientations to the lab for students and/or staff. A TA is on duty for 50 of the 60 hours the lab area is open, including all peak hours and Saturdays. Each TA is expected to become familiar with most of the available software and all of the hardware. Some tasks such as changing ribbons and print wheels on printers are performed only by a TA to avoid equipment damage.

A 27-page laboratory manual is available to users for $1.00. Continuously updated on a word processor to reflect new equipment and policy changes, it contains sections on regulations, hours of operation, phone numbers for reservations, a glossary of terms, detailed instructions with diagrams for using each type of printer and microcomputer, diskette care, and supplies available at the desk. This manual addresses problems frequently encountered by the TA's as they work with users.

Two types of open orientation are held about eight times per semester according to a posted schedule. Although reservations are required, the orientation is offered at different days and times for the convenience of users. The first orientation - Micro I - insures that participants learn to use all available lab equipment, including printers, disk drives and the Apple keyboard. Additional objectives include knowledge of reservation and check-in procedures, rules for use of the lab, awareness of five most common problems encountered in the lab, and ability to feel comfortable using lab facilities. This orientation is a prerequisite for Micro II, which provides an introductory survey of lab software. Featured in Micro II are word processing, utility programs, CAI and tutorials, data management, authoring systems, and other software of interest to the group attending. Both 90-minute orientations include at least 30 minutes of hands-on experience, including tutorials designed to introduce new users to the Apple. These orientations are scheduled at the same hour one week apart to insure that those who attend the first can attend the second. All users are required to attend the first orientation to insure their proficiency on lab equipment. A file record of all who have attended Micro I is used by check-out desk personnel to assure that all clients of the lab are proficient with the equipment. This policy protects equipment and has greatly reduced both the time clients waste in becoming proficient and the number of questions TA's must answer.

EVALUATION

An hourly survey was conducted from September 24 - 30th, 1983 to establish the categories of use of the lab, peak hours, and degree of saturation of the equipment. At hourly intervals each work station was checked, and its use category recorded. At that time there were 12 work stations and three printers. Results show that the lab was at 77% of saturation for the week, with a range of 59% on Friday to 94% on Tuesday. Word processing accounted for 57% of use, programming for 35% (22% BASIC and 13% other languages), and other uses (including review and evaluation of courseware) accounted for the remaining 8%. It is felt that these values may fluctuate from week to week as assignments fall due and classes shift emphasis to new material. However, it is clear that word processing accounts for a growing portion of lab use by our clients.

An evaluation form is distributed at the conclusion of Micro I and II to help staff improve the services we offer. Participants indicate the greatest satisfaction with objectives relating to lab introduction, operating rules and care of diskettes, with an average of 93% checking "about right" for the success of these objectives. However, only an average of 60% of participants in the Micro I orientation checked the same category of success for objectives relating to "specific use of the computer for my needs" and "theory of computer operation".

IMPLICATIONS

It seems that experience is the best teacher when it comes to setting up and providing for the smooth operation of a microcomputer laboratory. Ironically there seems to be a chronic shortage of reports in the literature to serve as guides for those faced with this task. The paper by Weinstock (1983) discusses efforts at the Illinois Institute of Technology under a grant from the National Science Foundation CAUSE program. His paper includes a sketch of lab layout, procedures for courseware selection, discussion of public relations and staffing, as well as projections for future improvements. Thus, it might serve as a useful supplement to this paper for institutions of higher learning. Knight and Kull (1983) report on results of a local grant to the Departments of Computer Services and Education at the University of New Hampshire.

With current equipment and software for educational uses of computers in such a state of flux, it is perhaps not surprising that few have had the time to sit down and write up a description of how their microcomputer lab came to be. However, such "roadmaps" are badly needed by all who would establish laboratories at any level. Mistakes in getting started can be quite costly, and tend to become permanent. Perhaps this paper may be of some help.

REFERENCES

Knight, Anne H. and Kull, Judith A. "Microcomputer Lab: A Focal Point for Cooperation". unpublished paper. 1983. inquiries to

Dr. Kull at Dept. of Education, Morrill Hall, University of New Hampshire, Durham, NH 03824.

Moursund, David. "Microcomputers Will Not Solve the Computers-in-Education Problem". AEDS Journal Vol. 13, No. 1, Fall 1979, pp. 31-39. also reprinted in Microcomputers in the Schools . Oryx Press. 1981. pp. 112-116.

Weinstock, Harold. "Establishing and Maintaining a Multimedia Learning Center: A Case History". Journal of College Science Teaching . Vol. 12, No. 3. Dec.-Jan. 1983. pp. 142-150.

Wells, Malcolm and Bitter, Gary. "The First Step in Utilizing Computers in Education: Preparing Computer Literate Teachers". Report to the National Science Foundation. 1982. ERIC Document ED 218 703.

SOFTWARE: COPYRIGHTING, PROTECTING, PIRATING

✓Robert K. Bane, Ed.D
&
Wayne P. Hresko, Ph.D
College of Education
North Texas State University

Abstract

At the same time that schools are acquiring ever increasing numbers of computers, another practice, that of software piracy, is mushrooming at an even faster rate. Apart from the ethical and legal questions involved, the consequences of this practice may very well be increased costs for software as well as reduction, if not the demise, of future educational software development. It is the responsibility of educators to work toward the curtailment of illegal duplication and distribution of software. The ICCE policy statement is suggested as one place to start in this endeavor.

The Problem-

Successful applications of computers in education will ultimately depend on the quality of the software that is developed to meet a wide variety of educational needs. At the same time that schools are acquiring ever increasing numbers of computers, another practice is mushrooming at an even faster rate that may very well spell the demise of educational software development. The practice we are referring to is software piracy. In this context, software piracy refers to illegal duplication and circulation of copyrighted software.

It is not particularly difficult to understand some of the reasons why software piracy exists within the schools. In some instantces it is simply a carryover into the schools of a practice which is rampant in the larger community. Among large segements of society piracy has become not only an acceptable practice, it has become an expected one as well. Another reason piracy has become so prevalent is that software can be very expensive and if inexpensive copies cannot be made for use with a large number of computers and students then the cost factor may become prohibited. Still another reason, one of a social nature rather than economic, is that the practice by teachers of sharing ideas and materials is a long standing tradition. The sharing of software whether copyrighted or not is simply a modern day technological extention of what has always been an accepted norm among teachers. Another reason is that copying software is very inexpensive and easy to do. For an investment of about $2.00 for a blank disk and about 2 minutes of time it is possible to obtain full use of a program that took thousands of hours and dollars to produce.

Just how widespread is this practice? Consider the following. One of the authors recently purchased a Commodore 64 computer. In less than one day he was offered in excess of $1500 dollars worth of illegal software! And, he never solicited any! If had chosen to do so, he could have obtained everything from a sophisticated word processing program to a data based management system to any one of dozens of published games. We also know of teachers who hold "swap" meetings with other teachers and students, where programs are traded without any concern for the ethics involved or the legality of doing so. We would contend that there are very few, in any, readers of this article who have not at one time or another been offered illegal copies of software.

Pirating has become so widespread in the schools that a pioneer in educational software development, the Minnesota Educational Computing Consortuim (MECC) has changed its policy so that all future products will be copy protected. For more than a decade MECC has been involved in the production of high quality, low cost educational software. School districts that wished to acquire their software could pay a licensing fee plus a small charge for each program disk (approximately $5.00). Member school districts were then permitted to make unlimited copies for distribution within their district. Though the cost to schools was well below that of commercially developed products, the revenues were sufficient to support continuing product development. MECC changed their policy when they became aware of the extent to which nonmember school districts were copying their programs.

Yes, software piracy is widespread and growing. It is rapidly becoming a problem of such magnitude that it may very well have dire consequences for computer related education.

The Consequences-

Certainly, one consequence will be an increase in software costs rather than a decrease. The MECC policy change is a prominent example, the result of which will be an increased expense for all school districts including those who met the law.

Second, and definitely the most damaging consequence, is that if the high rate of piracy in schools continues, there will be a significant decrease in the number of quality programs available. Software publishers are in business to make a profit. Why spend extensive development time, and initial outlay costs, with little chance of making a profit?

Third, when schools pirate software, they encourage their students, by example, to do the same. Thus, the lesson that students learn is that it is all right to steal the ideas and labor of another individual. Is this what we really want to model for our students?

A crucial issue is how we as educators can remedy this problem. But, perhaps the most crucial question of all is: "Do we really want to do something about it?

The Solution?-

It would be naive to think that there is a final solution that will totally solve the problem of software piracy. It does seem reasonable, however, that measures can be taken to reverse the tide, such that if piracy cannot be stopped completely, it can at least be curtailed.

A first step would be for all educators to acknowledge openly that making multliple copies of copyrighted software and sharing these programs is nothing more nor less than out-and-out theft. It is illegal, unethical, wrong, and injurious to the welfare of the creator of the program. We can do nothing less if we are not to become a walking contradiction of the trust that society has bestowed upon us to teach among other things, the virtues of honesty.

Fortunately, the problem is being addressed at the national level. In the September, 1983 issue of The Computing Teacher , the International Council for Computers in Education (ICCE) issued a policy statement aimed at fostering cooperative action among educators, hardware vendors, and software vendors and developers. The ICCE proposal covers several problem areas.

First, ICCE proposes that schools take an active role in monitoring the unlawful duplication of software within their environments. School based actions include enforcing copyright law and publisher license agreements, monitoring of school or district owned software, preventing unauthorized copying or use of copies on school equipment and educating students regarding the legality of copying software.

The second major aspect of the ICCE statement concerns the responsibilities of hardware vendors. ICCE suggest that hardware vendors assist both educators and software vendors and distributors by helping develop encryption processes which both allow for the needs of school districts while inhibiting illegal software duplication.

The third major section focuses on the responsibilities of software vendors and developers. Essentially, ICCE proposes the development of systems which ensure the provision of back-up copies for archival purposes, flexible encryption techniques, and flexible and discounted price structures for both multiple copies and network based systems.

The extensive nature of the ICCE statement precludes an indepth analysis; however, we would also mention that

included in the statement are suggested models for school district policy as well as college and university policy, and suggested formats for software licenses and encryption. By no means the final answer to the problem of piracy in educational settings, it does provide a framework from which meaningful solutions may emerge. We encourage all involved in computers and education to carefully review the ICCE policy statement.

In summary, the problem of software piracy within our schools is a rapidly growing problem which may have adverse consequences for the the future of educational software if measures are not taken to stem this practice. It would be a sad day, indeed, if sometime in the near future, one were to hear it said, "Say, whatever happened to educational software?"

TEACHERS AND COMPUTERS: WHAT KNOWLEDGE IS OF MOST WORTH?

Dr. Robert K. Bane, Judith Schneider, M. Janet Spence
School of Education, North Texas State University
Carolyn Pope
Desoto Independent School District
Lynne Rigg
Garland Independent School District

This presentation will focus on the knowledge that teachers need in order to use the computer effectively as a tool in the classroom. The teacher should be the facilitator for classroom computer use rather than a programming instructor. He/she must be competent in the daily operations of the hardware and must be a discriminate user of available software. The method of integration into the curriculum will depend upon whether the purposes of instruction are computer literacy or CAI usage. The teacher will realize the greatest advantages in computer usage when CMI packages and grade-reporting packages are available.

Teachers as Facilitators

According to U.S. Department of Education data, the number of computers in public schools has tripled in the past two years, and three-fifths of the nation's secondary schools now have them for student usage. The elementary classroom teacher, the junior high English instructor, and the high school biology teacher are examples of educators who are successfully implementing computers within their respective subject areas without benefit of an extensive computer science education in most instances.

The authors contend that teachers who fit into the aforementioned categories do not need to know how to program computers in order to function effectively as "facilitators" for their students. These teachers, by and large, are risk-taking, innovative leaders who are willing to expose their students to the computer world by plunging in and learning about computers right along with their pupils. With little in-service training and using the computer manual as their textbook, these teachers simply learn by doing.

Teachers who use computers as tools in the classroom are most often dealing with educational software in the form of cassettes or diskettes specifically related to the subject matter they teach. Programming capability is not required for this type of computer usage, and with a little help from a well-written manual or personal instruction from another teacher, software usage becomes as routine as writing on the chalkboard.

Once teacher "facilitators" become comfortable with the computer in the classroom, they often seem to catch "computer fever", i.e., the insatiable thirst to learn more about them. Many teachers who originally felt somewhat threatened by computers but were allowed to integrate them into their classrooms in a timely manner with minimum administrative

pressure, now feel comfortable with the hardware and want to learn to program.

After having tried various software packages, many instructors feel that commercially-prepared programs do not adequately address their specific classroom needs in many instances. The availability of authoring systems which will allow teachers and curriculum developers to create courseware presents a supreme challenge to educators who wish to expand their capabilities into this area. Although authoring packages are well-designed in most cases, teachers who plan to use them will soon discover that knowledge of even a minimum of programming language essentials will be a tremendous asset to them.

Depending upon the type of software they are using, teachers will be working within the confines of BASIC or LOGO languages. BASIC is the most popular language in use today for K-12 educational purposes. Although most elementary level software is written in BASIC, LOGO may also be used in some elementary programs.

Once teachers begin to learn a programming language, they will begin to write very simple programs on their own and progress into writing more complex programs as their expertise increases through experience coupled with additional training.

In summary, teachers who serve as "facilitators" in computer implementation and who are not directly responsible for teaching computer science as pure subject matter can be effective without knowing how to program. After using the computer in their classrooms for a period of time, many teachers will begin to see that a knowledge of programming is a definite asset and will WANT to learn at least simple programming techniques. It is this desire to learn which makes teachers the most effective in integrating computers into their classroom.

When teachers understand that it is not necessary to be programming experts or technicians to use computers in their classrooms, they feel less threatened and resistance to implementation of computers is likely to be less of a problem factor.

Operational and Software Knowledge

As more and more school districts employ the computer as an adjunct to the regular classroom teacher, the need for educators and administrators to become instant "experts" in software evaluation and usage is critical. The question of what the teacher needs to know in order to use the software effectively is one that must be addressed immediately.

There are three basic areas of knowledge that appear to be most necessary in preparing the teacher for a successful experience with software. One involves a basic knowledge of the hardware being used. The second requires an ability to evaluate the efficacy of particular software programs. The third and most basic area is an understanding of the content being presented, the prerequisite tasks required for the student to

use the software (both in terms of his/her knowledge of the computer itself and his/her knowledge of the concepts being presented), and a methodology for evaluating the student's progress in order to provide remediation or advancement.

A rudimentary knowledge of the hardware system provides the teacher with some ability to deal with the simple problems that often plague users. Statements like "syntax error", "disk full", "unknown command", "type incompatible" and so forth can be dealt with with a modicum of disruption. Should the hardware freeze-up or some other problem present itself, the teacher would have some trouble-shooting skills available.

The ability to evaluate software would enable the teacher to use those programs which best meet his/her own instructional needs. Since there are great variations in student capabilities and needs each semester, programs which were valid for use one semester may not be appropriate for use the next semester. The teacher should be able to evaluate whether or not the presented activity is appropriate to and integrated into the concepts to be taught and is appropriate to the age of the user. The called-for responses must be reasonable tasks for the user. The use of graphics and sound must enhance and support the purpose of the program. These criteria are only a few of those necessary for evaluating software. Teachers must be aware of the criteria and be able to evaluate the software accordingly.

While the computer may be used to provide the student with a rather simple approach to a particular concept, the teacher must have the ability to clarify, expand, and enhance the concept should this be necessary. Moreover, he/she should be able to discern the prerequisite tasks necessary for the student to use the software successfully. A student who is ill-prepared either in terms of his/her knowledge of the system hardware or in terms of his/her background in subject area will experience frustration and failure. Therefore, a knowledge of prerequisites is vital.

Educators have an obligation to become critical evaluators of the educational software that is currently being used and distributed. However, in order to do this effectively, they must have competence in the aforementioned areas.

Integration Into The Curriculum

If computer literacy is the teaching focus, integration with the regular classroom setting can be accomplished in one of two ways. The essential elements of literacy, except for programming, may be addressed in traditional program areas. For example, the history and development of computers could be included in the social studies curriculum while terminology could be addressed through language arts. Because of the need for hardware access and teacher expertise, the programming element would require special arrangements.

Another approach is to

address computer literacy as another subject for students. In this case, a separate curriculum and training program would be necessary for those who would teach computer literacy.

If computer-assisted instruction in the content areas is to be integrated with regular classroom instruction, other factors should be considered. Thorough staff development is the first step in this process of integration. To use CAI teachers need not spend precious in-service time learning about computers. Instead, this time should be devoted to study of the content and instructional design of the software. Then teachers can begin to develop correlations of the software to existing classroom materials. When the correlations are complete, the software can be infused with other resources in subject area curriculum guides and a central instructional management system.

All of this facilitates the real key to CAI integration--diagnosis and prescription via computer managed instruction. When students come away from a CAI lesson with an appropriate grouping and assignment to be addressed in the regular classroom, the process of integration is complete. The teacher is no longer simply managing instruction, he or she is freed to teach.

Possibly, the most critical elements for lasting integration of computer instruction with regular classroom instruction is the involvement and guidance of a key building level administrator. He or she also needs to be completely familiar with the content and focus of the instructional software. It is this person's responsibility to aid and insure that all teachers are using the aforementioned tools. Only with this kind of support will the overall program have continuity and consistency.

Planning and Record Keeping

A computer can make the teacher's daily planning faster and more efficient only if the system has a powerful computer managed instruction component. When the CMI software can analyze learning patterns and prescribe CAI and classroom instruction the computer is truly a teacher's tool. Without this CMI component, the teacher faces the monumental task of tracking the students' CAI performance by hand. This quickly becomes a nightmare as students begin to differentiate.

Grade reporting and home/school communications can be significantly easier when using a computer. Many software packages exist that make grade averaging and progress report writing things of the past. Not only is teacher time saved, but records tend to be more detailed and can be stored efficiently. Computing power is easily utilized for these tasks.

Summary

In summary, the authors feel that teachers can be effective in the implementation of computers in the classroom as a teaching tool without the benefit of programming

knowledge. While programming may be an asset, it is not a necessity for most classroom teachers. More importantly, they must be able to operate the hardware efficiently and react to routine error messages and problems. They must be able to evaluate the software which is available in terms of appropriateness of the content and the prerequisite skills needed on the part of the student. They must be aware of the purposes of computer usage. Finally, if available, the teacher can reap the benefits of being a computer user when CMI packages, grade reporting and word processing software packages are at his/her disposal.

Beginning a Computer Program

Josephine Basel
3000 Hillbrook
Dallas, Texas

Before buying software and hardware, districts need to determine their needs including curriculum requirements, population to be served, financing available, methods of staffing and facilities available.

Nationwide, districts are making computer technology one of their priorities. Here in Texas, computer literacy is being included as part of the curriculum being developed at the state level. Each district needs to begin developing a plan for implementing these requirements.

The first step in any good educational program is to determine the needs of the district. A curriculum needs to be decided on that sets the standards for what will be taught at each level, and what level of usage is to be reached. If the district has not yet decided on a curriculum, a proposal is available from the Texas Education Agency that will give guidelines for establishing a program. At the time of establishing a curriculum, it should be decided who will be responsible for implementing this curriculum.

Facilities need to be surveyed for the existence of appropriate space for establishing a program and the suitability of these spaces. Some factors that need to be considered are accessibility to students and faculty, availability of electrical outlets, lighting conditions and enough space for equipment required. At this point, a determination

needs to be made as to what kind of situation the school or district desires for computer use. There are basically two methods of setting up computers. The first would be in a lab situation that would require a classroom or to have the computers distributed through individual classrooms. Much of this would be decided by the curriculum established in the district.

Once it is determined that facilities are available, types of equipment should be decided. Looking back at the curriculum that has been decided on, a match needs to be made between the curriculum and the capabilities of the equipment purchased. This decision cannot be made without looking at the financing available. At the outset, there are three sources of funding - local, state, and federal money. In addition, there is the possibility of private funding from business and industry. If the school has a parent - teacher organization that helps with financial support, this is another possibility. The possibility of there already being equipment existing in the district needs to be investigated and an inventory of what is already there made. Uses of the computers besides that of teaching computer literacy should be considered. There is also a possibility of using these same computers for computer assisted instruction and also for management problems.

At the same time equipment decisions are being made, personnel that are to be teaching with computers should begin their training. These same people should have been involved in the curriculum writing process, so that they have knowledge of the development of the whole plan. Without total involvement and enthusiasm on the part of personnel, the computer program will have a difficult time succeeding.

Once equipment is purchased and initial personnel requirements are met, a systemwide training process

for teachers and administrators needs to be implemented. This plan needs to include training in basic computer literacy as well as use of the specific equipment purchased by the district. Work with correlating instruction with computer use should be done in these sessions.

Although one of the determining factors in buying equipment should have been the amount of software available, more serious consideration should be given to the purchasing of software at this point. For the teaching of computer literacy, only limited amounts of software need to be considered, but if computer assisted instruction or management is to be done either more software will have to be purchased or people with the ability to program will have to be found. Teachers who will be using computers with their students or who will be sending students to use computers need to be aware of what kind of software is available and how to correlate that software with their instruction. Faculty need to be involved in the review and evaluation process and those working at the district level need to make sure that each piece of software is carefully evaluated for content, applicability to curriculum and use on equipment.

In summary, the following steps need to be followed in beginning a computer program:

(1) Objectives need to be set.

(2) Curriculum needs to be written including standards to be met and levels of population to be served.

(3) Facilities need to be investigated.

(4) Financing needs to be found.

(5) Number of population to be served needs to be decided on.

(6) Amount and type of hardware and software needs to be decided on.

(7) Number of personnel needed to meet goals needs to be determined.

(8) Training needs to be made available at all steps of the process.

Networking with the TRS-80

Josephine R. Basel
3000 Hillbrook
Dallas, TX 75214

Networking could be an effective solution to the shortage of funds in some districts for establishing a computer program. In managed properly, it can also work to a greater educational advantage. One company selling networking equipment is Radio Shack.

A network is defined as any system of lines that cross. In radio or television a network is a group of stations working together, so that what is broadcast by one may be broad by all. Networking with computers operates exactly the same way with data that is being used by one computer able to be communicated to other computers.

Radio Shack now has available two versions of a newtwork controller. These are the network 2 controller and the network 3 controller. Each of these controllers can utilize up to 16 student stations with a host (master computer) to operate them. The network 2 controller can operate off either a disk drive host or a tape driven one. A cable connection is used to connect host to student stations. A network 2 host must have at least 48k of memory. In the network 3 system, the host must have at least 64k of memory and each student station 32k of memory. The network 3 connecting cables use an RS-232 interface.

The actual process of networking with a network 2 controller is begun by loading a program into the host computer. After the host is loaded and ready each student station is brought up ready to

load. At this point, the controller is used to download the program from the host to the student stations. With the network 3 controller, the process is not a download, but each student station has direct access to the information on the host.

There is software available for networking, but the purchaser of software for a networking system needs to be aware of possible problems. In order to use software on a networking system the program has to be unprotected or the program cannot be protected against copying. Most companies are currently charging twice the list price for software that will network and some are asking that the purchaser sign an agreement that states no copies will be made except for use on the purchaser's system. In addition, there are differences in network 2 and network 3 software. Network 3 software wil' not load on a network 2 system, because it is not set up for the downloading process. In addition, if network 3 software is purchased, a management system is available with some pieces that will allow the teacher to keep track of a student or students and put the students back into the program where they exited it. At present, the best two sources encountered for networking software are Radio Shack and Random House. Both give assistance in obtaining appropriate software.

The best way to use the networking system would be in a lab situation with the host, controller, student stations, and a printer available for student results. The lab needs to be staffed with a person capable of both teaching with and about computers and also capable of handling minor technical problems. In an elementary, middle school, or high school a lab could operate as an adjunct to the learning resources center. This would work best if the principal uses were to be for teaching computer literacy or for use in computer assisted instruction. If

advanced programming was the goal or use in word processing, an isolated classroom staffed by certified teachers would probably be a better idea. It is possible to teach computer literacy without the computers available, but as in most cases, hands on experience is one of the best teachers. Computer assisted instruction is applicable in many areas including math, language arts, social studies, reading, science and foreign langauges. In addition, with the technology that is now coming, work in the fine arts can also be done with music and graphic arts. CAI can be used with all levels from special education students to the gifted and talented, from early childhood education through college.

The advantages to networking are threefold:

(1) Only one set of software is necessary.

(2) Only one computer must have some kind of drive; the student stations operate off of that one computer.

(3) The instructor does not have to spend time loading or assisting in loading each computer, but can still monitor the placement of each student.

In addition to these advantages, there are others. Less software is lost through multiple use. Since networking software is unprotected, the original may be copied and then saved for the time that the copy may be damaged. In those software programs where a management system is available, the teacher may keep records of each student and class.

There could be possible problems with scheduling. In districts where computer literacy and programming have been made a part of the curriculum, time may be scheduled for these subjects in the junior and high schools. At the elementary level, scheduling is more difficult because the day is full of assigned instruction. Computer literacy could probably best be made a part of the math curriculum and scheduled as

regular instruction. If the computer lab were adjacent to the learning resources center, classes could be scheduled into the lab with their instructor just as they are scheduled for times in the library or other situations.

Given the budget that many school systems are trying to begin their computer programs with, networking could be the answer for many with financial savings in hardware, software and staffing.

Staff Development

Josephine R. Basel
3000 Hillbrook
Dallas, TX

In some areas, there are requirements for computer course being instituted in teacher education programs. There is also a need for educating teachers currently in the program. Planning for this needs to be a step-by-step process.

In these days of the development of computer literacy and technology for many school districts it seems that teacher education colleges are going to require that each teacher education candidate take a course in computer use similar to the audiovisual course that they were once required to take. The implications of computer use are going to be much farther reaching than those of audiovisual equipment. It is to be hoped that this course will be taken seriously by both teachers and students.

It is not going to be enough to just educate those teachers coming into the field, but those teachers already teaching must be instructed in the use of computers and implications for their students and themselves. Several steps are needed to fully implement a plan of staff development for computer literacy and computer assisted instruction. Stages possibly necessary in a district plan are as follows:

(1) Overview for all teachers in the district to last approximately 3 - 4 hours. This could be done by a someone hired from an outside source or by a specialist in the district.

(2) A computer literacy course required

for all teachers to include use of equipment to be used in the district. This should last approximately 10 - 20 hours and be taught by personnel from the district. An alternative could be a course offered for college credit to be added to the teacher's service record.

(3) For those teachers who will be directly involved in teaching with computers either with their class or schoolwide, an additional course should be required in implementing the curriculum combined with computer use to include correlating classroom instruction with computer instruction. This should last approximately 10 - 20 hours. This would be another case where alternative instruction could be offered for college credit. This should also include those teachers involved in teaching computer literacy.

(4) Higher level instruction may be necessary to obtain teachers who can teach programming at the middle and high school level if these teachers continue to be in short supply. The same thing may be necessary for word processing skills. Both of these courses would probably need to be taught outside of the regular district program with possible college credit or work experience required similar to that necessary for vocational education.

The attitude of most teachers at the moment seems to be that the computer program exists outside or external to the regular classroom. If the teaching profession is to succeed in teaching computer use and literacy those objectives and skills need to be reinforced within the regular classroom. If one of the reasons for teaching computer literacy is that computers are a force in everyone's life today, whether good or bad, then teachers need to help students determine the good and bad uses and also help them to maintain a balance so that the use of the computer does not take the place of all other learning.

In an effort to maintain a balance between using the computer and

correlating it with regular classroom instruction, the following plan was developed as a possible strategy to use.

Upon examination of the ITBS scores in the fall, it was determined that School X had 30% of its students score below the fiftieth percentile in reading. A schedule was arranged so that those students would go to the computer lab with an instructor and work on specific reading skills as determined by the district. A computer program is selected that targets reading skills. Each student is given a contract which offers specific paper and pencil activities to remediate the students' specific deficiencies as determined by the instructor. After completing the activities, the student is put onto the computer at a lesson appropriate for his needs. After successful completion of the computer program, the student is again reinforced with paper and pencil activities.

If the classroom teacher can be made aware of the effective use of the computer and can correlate this use with classroom instruction, the computer can be one of the most efficient and effective teaching tools available.

"Input: Planning and Staff Development. Output: A Successful K-12 Computer Curriculum"

✓Julia Berry and Pat Kramer
Frisco I.S.D., Box 910, Frisco, TX. 75034

The strategic long-range planning necessary for implementation of a K-12 computer curriculum and the goals and objectives pertinent for the design of such a curriculum will be presented. An effective staff development model and the Frisco I.S.D. computer curriculum guide will also be presented.

Extensive planning and staff development are of utmost importance for implementing a successful K-12 computer curriculum. The first stage of planning should be to administer a needs assessment to staff, parents, and students in order to determine awareness, experience, and interest in computers. Suggestions for computer usage should also be solicited.

The next step in planning for computer usage in the schools should be the formation of a task force comprised of administration, parents, students, staff, school board members, and community members. A conscious commitment should be made to the integration of computers into the curriculum. The task force should seek out all available resources and literature, visit schools using computers, attend workshops, and talk to computer-literate professionals and children. The task force should also consider possible problems in the implementation of computers in the schools.

The next important stage in planning is to formulate goals and objectives for computer usage. Will computer assisted instruction be the primary use? What subject areas and grade levels will be involved? In what languages will programming be taught? Will computer managed instruction be utilized? What hardware best meets the goals and objectives?

Perhaps the most vital step in incorporating computers into the curriculum is inservice training. The staff must be aware of the objectives for computers in the school program. An effective staff development model for the Frisco Independent School District has incorporated the following topics into an eight day summer computer "boot camp":

1) Immediate hands-on experience with the computer, including an introduction to the keyboard and the capabilities of the computer
2) Computer related terminology
3) The history of computers
4) Practical application of the computer for daily classroom uses
5) Copying a program
6) Using various peripherals
7) Software evaluation
8) Social implications of the computer
9) Computer careers

10) Computer support materials
11) LOGO
12) Superpilot
13) Previewing software

This workshop has been conducted several times both within and outside district, and was evaluated very highly. The usage of LOGO gave staff members immediate success in programming and a feeling of "control" over the computer. Other ongoing staff development includes word processing and other programming languages.

After developing a core of trained personnel, extensive planning, and some experimentation, the district is prepared to write a computer curriculum.

Implementing the Curriculum

To meet the demands of a highly technological informational society, the follwoing 6 areas should be presented to students, regardless of grade level: 1) Computer-related terminology and use 2) History and development of computers 3) Use of computer as a tool 4) Communicating instructions to the computer 5) Social implications 6) Robotics

At the elementary level, students accomplish these objectives during their scheduled library periods via computer literacy lessons, CAI, and LOGO programming. Middle school and high school levels acquire these skills within the social studies classes through CAI, computer literacy lessons, LOGO programming, encompassing higher level thinking skills and problem solving, and social implications. The high school students also are offered BASIC and PASCAL as electives.

Since this is a new curriculum area, adjustments in the introduction of skills will be made as prerequisite knowledge is attained. Tentative plans are being formulated to move BASIC instruction to the middle school level and to add another higher level language to the high school curriculum. The curriculum will be modified to keep abreast of swiftly changing developments in the computer field. Naturally, ongoing staff development will be imperative to the success of any and all computer curriculum undertakings. Just as the curriculum will broaden so too will expand the teacher training objectives.

(See A Framework for Educational Computer Usage, K-12, published by the Frisco Independent School District, 1983, for the actual curricular framework).

INTERCHANGING DATA AND PROGRAMS BETWEEN MICROCOMPUTERS AND MAINFRAMES

Pearl W. Brazier
Pan American University
Edinburg, Texas 78539

Methods to exchange files between microcomputers and mainframes are discussed. Uploading a file from a microcomputer is described and some of the problems are discussed. Downloading a file from the mainframe to be printed on the microcomputer or save on the floppy disk is described and discussed.

I. Introduction

Microcomputers are becoming more available and are being used for processing in environments where an existing mainframe or mini-computer serves some of the computing needs. While much computing power is available on the stand-alone microcomputer, there are situations where it is desirable to be able to share data and programs with the larger machine. A mainframe or mini-computer may contain a database of student records or have more efficient utilities such as sorting and merging available. The mainframe may also have hardware resources, such as printers, that could be used in certain situations. A limited number of terminals or a limited amount of access time may be available on the mainframe, and in this case the micro can be used to enter data and programs to be stored in the micro's disk to be sent to the mainframe at a later time. The mainframe could also serve as a communication link between two microcomputers from different vendors. Data can be sent to the mainframe from one microcomputer and then sent from the mainframe to a different microcomputer, thus bypassing the problem of incompatibility between different brands.

This presentation will descripe programs that can be used to send and receive text files from the microcomputer to a mainframe.

II. Equipment

The programs to be described are designed to run on the Apple II Plus or the Apple IIe equipped with a Hayes Micromodem II. The author modified several programs that were supplied with the modem. One of the original programs was designed to extract a BASIC program from a mainframe and the other program was designed to send text files from one Apple to another Apple over the phone line. There are also commercially available software packages that will allow similar transfer of data files.

III. General Principles

Extracting a file from the mainframe will be referred to as downloading a file. Sending a file stored on the micro's disk to the mainframe will be refered to as uploading the file. To do either, the Apple must have a program running which sends or receives characters, and the mainframe must have a corresponding program to receive or send characters to the Apple.

In order to upload a file, a text editing program in input mode can be used on the mainframe to receive the text. A system command to type the contents of a file from the mainframe can be used to download. The Apple must first be placed in terminal mode, either using an Applesoft program or using the control sequences described in the instructions supplied with the micromodem.

IV. Uploading to the mainframe

In order to transfer a file to the mainframe's disk storage, the text editior program on the mainframe is placed in the input mode. Control is then returned to the Apple, using the control keys, as indicated in the micromodem manual. The phone connection is retained, but the terminal mode is temporarily suspended. The upload program, written in Applesoft BASIC, is then run. This program asks the user for the file name to be uploaded, then opens and reads the file sending the characters to the micromodem output slot. When the end of file is reached, the program returns the Apple to terminal mode. The mainframe is still executing the text editor and the user can now save or edit the file on the mainframe.

The major problem with interfacing the two systems is timing. When the command is sent to the mainframe to begin typing out the file, the Apple must wait to begin receiving until the mainframe has received the entire command. The micromodem issues a RETURN to indicate when the command has been sent. The program needs a loop that continually checks for that RETURN before continuing.

V. Downloading

Once in terminal mode, a disk can be downloaded to the printer or to the disk connected to the Apple. Again terminal mode is temporarily suspended using the usual control characters and the Apple download program is run. This program requests the name of the file on the mainframe and issues the command to type the file. The program then accepts the text being typed, with INPUT statements storing the text in a string array. The contents of the array can then be output to either the printer or to the Apple disk drive. In order to handle large files, without using all of the Apple's memory, the file is typed and saved in small sections. The process is complete when the program detects the prompt character from the mainframe, indicating monitor level or the end of the type command response. At this point the program returns the Apple to terminal mode and the user can log off or continue to interact with the mainframe.

The primary problem encountered with this phase was one of timing. When the section to be

downloaded was large (more than 20 lines) some of the data was lost. One solution was to download smaller chuncks of the file, which automatically caused a delay in the transmission of the data. Another solution is based on the format of the file. The program used the Applesoft GET command to accept characters, and then formed a string which was output to the disk, thus allowing strings of any length. If the file is blocked, or in other words has return characters at intervals of less than 256 characters, then using the INPUT command rather than a GET caused the Apple program to execute faster and eliminated the loss of data. The 256 limit is the maximum string length for the Applesoft INPUT statement.

VII. Summary

When both a mainframe and microcomputers are available, the resources of both machines can be utilized. Being able to transfer files from one machine to another can enhance the total computing power of the organization. This presentation described programs to upload and download text files between a mainframe and an Apple equipped with a modem.

TEACHING PASCAL: WHO, WHAT, WHEN, WHERE, HOW ... AND WHY?

Dr. Douglas L. Brown
Department of Technology
Houston Independent School District
5300 San Felipe
Houston, TX 77056

This presentation will discuss some of the issues surrounding the use of Pascal in the secondary curriculum.. We will discuss what Pascal is and why it is important to include Pascal within the curriculum. We will also discuss what training and hardware is required to teach Pascal. There will be a brief summary of pre-Pascal materials which are available to introduce Pascal-like concepts to pre-secondary students. The presentation will conclude with a review of the curriculum proposed for the Advanced Placement Computer Science course and the role of Pascal in that course.

I. What is Pascal?

Pascal is a programming language which was defined by Niklaus Wirth in 1974. It includes concepts derived from previous languages, and it is closely related to ALGOL-60. Pascal was specifically designed to be a language used to teach programming skills. It incorporates features which encourage good programming practices and which permit easy debugging and maintenance of programs. Pascal also allows the construction of data structures which are much more difficult in other languages (such as BASIC).

Pascal includes a number of constructs which are familiar to BASIC programmers. These include 'for' loops, 'while' loops, and 'if.. then.. else' statements. Pascal includes additional repetition and conditional statements; this lets the programmer select the construct which most naturally corresponds to the program logic. Pascal contains a simple output statement (writeln) which is similar to the BASIC "PRINT" statement. A very short Pascal program is seen in Figure 1.

Pascal encourages the use of basic structured programming principles. A large problem can be broken into a series of smaller tasks, and separate program components (called procedures) can be written to solve each small task. Since each procedure can be written

```
program demo (input, output);
var
  first  : integer;
  second : integer;
  sum    : integer;

(*  This starts the actual  *)
(*  program instructions.   *)

begin
   first := 10;
   second := 15;
   sum := first + second;

   writeln ('Answer is', sum);
end.
```

FIGURE 1. Short Pascal program

and debugged independently, the large programming problem can be reduced to more manageable smaller problems. The smaller procedures are relatively easy to write, debug, and maintain, and very powerful Pascal programs can be assembled from these small components.

Pascal has been implemented on most microcomputer systems. The most common version is UCSD Pascal, a package which includes an editor, an assembler, a file manipulation utility, and a Pascal compiler. Programs are entered using the editor, and then they are compiled and executed. A beginner faces two tasks --- learning to operate the components of the Pascal environment (the editor, etc.) and learning to use the language itself. There are advantages to this system, however. Once the UCSD system is mastered, its use is identical on all systems, and programs written on one system can be run on all systems.

II. Why teach Pascal?

Pascal is becoming more and more popular, and there are a number of reasons why it should be taught.

First, Pascal encourages good programming practices. It encourages students to use a logical, structured approach to problem-solving. It encourages the construction of self-documenting programs; a good Pascal program can be "read" and understood quickly by any other programmer. It encourages students to carefully think through the solution to a problem before writing code.

Second, Pascal gives students access to other modern languages. Other modern languages (like C, PL/I, Modula, and Ada) are very similar to Pascal. Each language has its own special characteristics and advantages, but a good Pascal programmer can learn any other modern language quickly. Pascal is the entry-level language to these other languages.

Third, Pascal is the language of academic computing. If students intend to major in computer science in college, they will be expected to either know Pascal or to learn it as their first task. Languages like COBOL and FORTRAN are certainly taught and have distinct niches, but Pascal is the language which computer science uses as a foundation.

Fourth, Pascal should be taught to give college-bound students advanced placement credit. In 1983 ETS established a curriculum for the new Advanced Placement Computer Science course. This curriculum is based on Pascal programming and uses Pascal to explore sophisticated concepts in computer science. The AP Computer Science test will be administered for the first time in the spring of 1984, and Pascal programming skills will be essential for students taking this test.

Fifth, Pascal provides a tool for writing significant programs. Languages like BASIC can be used to write short (<100 lines) programs,

but BASIC programs become increasingly clumsy as they lengthen; it's hard to write a real-world program in BASIC. A well-written Pascal program may number thousands of lines, and the logic of the program should still be clear. As a program lengthens and becomes more complex, Pascal (and other modular languages) have a greater and greater advantage.

III. Who should teach Pascal?

Many teachers currently teaching other languages could teach Pascal relatively quickly. If a teacher is very familiar with any other language, Pascal will be relatively easy to learn. Some Pascal concepts (like 'for' loops and conditional statements) are common to any programming language, and the syntax is quite familiar. (Figure 2 demonstrates a typical 'for' loop in Pascal.)

Other concepts will be more or less familiar, depending on the teacher's specific background. Pascal's use of procedures and parameters may be familiar to Logo programmers, but BASIC programmers may be a little less comfortable with them. Some concepts, like the use of pointers and variable scoping, will be new to both BASIC and Logo programmers.

```
for index := 1 to 10 do

  begin
     square := index * index;
     writeln (square);
  end;
```

FIGURE 2. Pascal 'for' loop.

The amount of teacher preparation which will be required to teach Pascal will vary. If possible, a college-level Pascal course is very desirable. Workshops containing 20-40 hours of training will provide enough fundamental background to teach the language, provided that classroom work is supplemented with extensive programming problems. It is difficult to really understand a language without actually writing significant programs in it.

To teach the Advanced Placement Computer Science course, college-level training is essential. The ETS curriculum includes Pascal programming, but it goes well beyond that. It uses Pascal to explore the concepts usually covered in a college-level Data Structures course. A teacher will need the equivalent of that course as preparation before attempting to teach the AP course.

IV. When, Where, and How to Teach Pascal?

Pascal will normally be taught at the secondary level. At this time Pascal is taught after the student has had training in other languages, notably BASIC and/or Logo. Both languages introduce important concepts and are potentially useful. Logo introduces the idea of modular programming and procedures, while the syntax of BASIC is somewhat closer to that used by Pascal.

It is possible to teach Pascal to students without previous computer experience. One alternative is the use of pre-Pascal materials which are currently available. These include "Karel the Robot" and "Mazetalk". In both of these packages, students are taught the use of simple, Pascal-like languages as a prelude to Pascal. A second alternative is the use of a package like McGraw-Hill's "Computer Power". This package is a very complete one-semester curriculum which assumes no previous student programming experience.

Beyond curricular issues, other decisions will be important when you consider offering a course in Pascal. Pascal programming is hardware-intensive, and you must be certain that enough dedicated hardware is available. The amount of computer time which will be required will vary with the curriculum. "Computer Power" outlines a method by which one computer can support fifteen students. Other curricula, such as the AP Computer Science course, may demand much greater access.

It will also be necessary to acquire the necessary Pascal software. No microcomputer currently offers Pascal as a built-in language, so a Pascal compiler must be purchased. Depending on your system, this compiler might be UCSD Pascal (which is marketed as Apple Pascal for the Apple II microcomputer), Pascal MT, Turbo-Pascal, or some other alternative. The price for each system will vary, reflecting the features included in the package.

V. Additional Reading.

"Why Pascal?", K. Jensen, EDU, Fall, 1979, Digital Equipment Corporation.

"Apple Pascal: Hands On Approach", A. Luehrmann and H. Peckham, McGraw Hill, 1981.

"Oh! Pascal", D. Cooper and M. Clancy, Norton, 1982.

"Karel the Robot", R. Pattis, Wiley, 1981.

"Pascal Discoveries", S. Krute, Creative Publications, 1983.

"Mazetalk Discoveries", J. Goodnow, Creative Pub., 1984.

A COMPLETE COMPUTER LITERACY COURSE PRESENTED via CMIVI

HELEN TEMPLE BUCHAN P.O. Box 273 Norcross, GA 30091
ANANKE, INC.

The Computer Managed Interactive Video Instruction (CMIVI) course for computer literacy is available for teacher and student education. The course is designed around 153 instructional objectives and is based on a programmed instruction program which allows for reinforcement and repetition until the concepts are mastered.

Computer Managed Interactive Video Instruction (CMIVI) presents professional level standardized instruction via video tape or video disc as the teacher's presentation tool for the core content of a course. Interactive Video was introduced to education as a one-on-one tutorial instruction technique. The classroom-experienced educators at Ananke, Inc. feel that all instructional computer programs must be interpreted and applied to the individual needs of the student by a teacher before the learning will be meaningful or useful. The audio of the tape or disc must be highly structured and 100% correct. The video must be interesting and plausible.

The Computer Literacy unit in the CMIVI format is designed for classroom instruction so that the hardware and courseware costs are within reasonable limits. The twelve interactive video lessons are supplemented with an additional 18 lessons which include follow-up activities, evaluations, and individualized support for introductory programming.

Few teachers are trained to teach a course on Computer Literacy. CMIVI allows the teacher to go through a training seminar with the same interactive lessons that will be used with the students or to guide the students through the unit while learning the specific information. Never is a teacher left wondering what to do next.

Computer Managed Interactive Video Instruction includes video that is presented in 'bite size' learning modules with reinforcing text questions and statements presented between each module to evaluate the learner's comprehension of the specific information presented within context, to direct the teacher-led discussion, and to reinforce the learning of key concepts that need to be remembered at the end of the lesson.

Each question is computer managed in pretest, interactive, and posttest formats. Teachers can administer the pretest and interactive lesson to the entire class at one time with the use of scan forms which quickly produce individualized diagnostic information at the end of the session.

The posttest should be taken in the individualized mode since it offers the tutorial session in which the student is told how the incorrect response relates to the questioned concept and is returned to the same question until the correct response is selected and reinforced. The students are individually directed through the hands-on experiences with the computers in a ratio of three students to one computer by line for line instructions on what to do and clear explanations on why the procedure works.

The CMIVI unit on Computer Literacy is comprehensive in that the teacher who assists the students in the application and assimilitation of the information does not need to be a computer expert. A textbook may certainly be used to supplement the CMIVI unit, but the unit will stand alone.

The teacher's guide provides a summary of each audio/visual presentation so that the teacher can preview the material prior to the interactive session with the students. Each question used to reinforce the learning of key concepts and facts is written into a behaviorially stated objective for the teacher so that the correct response is provided within context to enable the teacher to discuss the information in a knowledegable manner. The activity sheets assigned to the students for follow-up are included in the teacher's guide with the correct answers or suggestions for use.

The course is designed around 153 instructional objectives which are divided into eight strands including:

I. 283 vocabulary terms
II. Computer Operations
III. Computer Applications
IV. History and Development
V. Logic and Problem Solving
VI. Concepts of Programming
VII. Social Impact of Computers
VIII. Career Opportunities

The vocabulary terms are systematically presented, instructed, and reinforced to allow the student to have proper exposure to each difficult concept before it is questioned for evaluation of learning. The teacher is provided a chart which shows in which lessons each word is used and the level of use for each term within each lesson. The chart guides the teacher in reinforcing the vocabulary and to use available materials to supplement the instruction at appropriate times.

Each of the 161 interactive questions are coded to the 153 instructional objectives to provide diagnostic information on individual student strengths and weaknesses for separate lessons and cumulative reports for individual students and classes. An additional 120 questions are asked in pencil/paper format to provide review, to reinforce conversational use of the vocabulary terms, and to evaluate long-term retention of the major concepts. The degree of progress attained by each student and each class is evaluated by comparing the results of a comprehensive test given at the beginning and at the end of the course.

A career chart is kept by the student throughout the unit with most of the career information presented within the student's notebook. The chart enables the student to evaluate various environmental, educational, and social factors related to a possible career choice related to computers. Typical illustrations of the careers are included in the video material. At the end of the unit, a comprehensive test is given on the 16 entry-level careers to summarize the information and evaluate the student's learning of the key career-related concepts.

During the two history lessons, the student will complete a summarized chart which clearly outlines the historical time-line statistics related to the key developments which led to present day computer theory and technology.

The Computer Literacy unit includes 12 video tapes/discs, 12 diskettes, one teacher's guide, a reproducible student's book, a copyable diskette to be used with the student's book for hands-on instruction on programming procedures, and an interface card that allows a computer to interface with an industrial grade Video Cassette Player or Video Disc Player.

Our CMIVI requires one interactive hardware system for each teacher of Computer Literacy within a building. A ratio of one smaller computer per three students in any one class is recommended for lab procedures with the copyable diskette and individualized posttests.

Cost effective? One CMIVI system can teach the teacher and up to 30 students per class, five classes per day. An investment of about $9,000 allows one to teach 150 students per course duration for at least five years. Even better, you will no longer need to pay to teach teachers about computers only to lose them to industry.

A CMIVI Health Unit was used during the 1982-1983 school year as a 12 week course under professional independent administration. The results show that the 450+ students instructed via CMIVI showed improvement at least 250% higher than the improvement of the similar sized control group based on a comprehensive pretest/posttest comparison.

Summary: A COURSE DESIGNED FOR COMPUTER LITERACY

DEFINITION:
A THIRTY LESSON COURSE designed for students, teachers, and administrators who want to become literate about the world of computers.

The course is designed for use beginning at the 8th grade level. Since it has a very high information density, it can be used through the sophomore level in college and for teacher education until initial computer literacy needs are met.

CMIVI is a complete instructional package. CMIVI combines video tapes or discs, computer managed programmed instruction, and correlated printed materials for both the students and the teacher. CMIVI provides the lesson plan, course objectives, vocabulary, instruction, interactive questions, objectives for each question, pre/posttest, and instructional techniques necessary for effective comprehensive instruction. The programmed instruction format assures that every objective is taught. Completely coordinated text documentation for teachers and students is included to support both instructional enrichment and efficient learning.

CMIVI requires NO KNOWLEDGE OF PROGRAMMING. If you can read and turn on the computer, you can use CMIVI. The teaching of the Computer Literacy unit with CMIVI requires very little knowledge of computer literacy CMIVI teaches the teacher either before or while teaching the students

Implementation Of A Computer Literacy Unit For Eighth Grade Middle School Students

By
Harry Charles Port Neches Middle School, Port Neches, TX 77651

And
Dr. Sandra Haven Lamar University, Beaumont, TX 77710

The Port Neches ISD inaugurated an Introduction to Computer Literacy unit for all eighth grade-middle school students in the Fall of the 1983-84 school year. A curriculum and workbook had been written the previous summer for a teacher computer literacy course; and this, in modified form, was used as the basis for the three-week eighth grade course which was to be taught in the earth science classes. The four eighth grade science teachers (only one of whom had computer experience) were given an extensive three-day course of instruction in preparation for the projected task.

This article discusses the implementation and the results.

Port Neches is a suburban city in an industrialized section of southeast Texas. The school system has approximately 5000 scholastics who are from a white upper-middle class background. The school's curriculum is progressive with emphasis on basics, provisions for students at all academic levels of abilities, and a wide range of electives and extracurricular activities. During the 1982-83 school year a Gifted/Talented program was begun K-8, and one TRS-80 Model III single disk drive was placed on each of eight campuses below the high school level. Additionally, an APPLE II was placed in each middle school. All these systems were to be used with the G/T program, called PRIDE.

The PRIDE teachers, one on each campus, had been given a 3-day seminar on computer use with selected software. Naturally, it was only a short time before the PRIDE students and teachers began branching out into beginning BASIC programming. In the middle schools an on-going unit was written for the students by one PRIDE teacher; and before the 1982-83 school year ended, each PRIDE student had written, entered, and saved on a diskette one graphics program each on the TRS and APPLE as well as several other shorter programs.

The unit had been successful but had only reached a small percentage of the middle school eighth grade population. Another school program directed toward attaining computer literacy for teachers had been conducted within the school district during the same year. It was decided to extend and possibly incorporate the best features of both these endeavors. In the summer of 1983 one PRIDE teacher wrote a 15-lesson book and course of study to be used in a 3-week hands-on unit for computer literacy. This was to be part of the curriculum for each eighth grader and was to be conducted in the science classes. In expanded form the same book was to be utilized in the continuing teacher training program.

Each of the 15 lessons in the book was designed around some particular computer concept and consisted of an explanatory section, exercises on that con-

cept, and extra activities to accommodate students who either worked faster or who had a prior knowledge of computers. The overall objective was "to develop a fundamental understanding of and familiarity with the computer and its capabilities (and limitations) so we can see how it can be constructively used to aid us in our work, school, home, society, and recreation. Also (and not incidentally), we will strive to develop the skills necessary to read and write computer programs." Two commercial programs and overhead transparencies were purchased as were posters about computers and a SHOW AND TELL kit. These were augumented with other demonstration programs and overhead transparencies specifically prepared for this course. Sixteen TRS-80 Model III's were set aside from the school's holdings to be used in the course. Fifteen were intended for student use and one host system was to be used by the teacher as a monitor with a Network II. The demonstration program would be piped into the student's computers.

A computer lab was set up at each middle school campus and equipped with the necessary furniture and an alarm system. Since the PRIDE teacher who had written the book was on one of the middle school campuses, the course was to begin there. The school year is divided into six six-week grading periods and the first round of classes began at the first of the second six weeks. Fortunately, the eighth grade science classes in this school were taught at different periods, so the two teachers could alternate using the computer lab. This worked admirably. The lab was kept on campus for four rather than three weeks so that make-up time could be had with the systems. Then the lab was transferred to the other middle school campus. Here, a scheduling conflict because of everlapping schedules necessitated that the lab be set up for eight weeks. During the school day the computers all received active use. At night the class for teachers continued, first at one campus and then at the other. In the meantime a Part II of the teachers' course had been written, and 400 participants have taken one or both of the courses offered since the spring of 1983. In the spring of 1984, a Part III course will be offered to the teachers.

The initial decisions had been formulated and the physical hardware and book acquired. Next came the details of implementation which had to be settled. There were 440 students pre-registered for eighth grade for the 1983-84 school year. They were roughly divided evenly between the two middle schools. Since the classes each ranged in size from 23 and 29 students it was decided that 15 systems for student use would suffice if the students were paired on each computer. This, of course, led to many questions of management. In August, 1983, the four eighth grade science teachers were "walked through" the course during the intensive three-day work session. Only one of the teachers had ever received any formal training with computers. A list of questions had been prepared for a teacher's manual which was considered and tentatively resolved during these sessions. Some of these questions were:

Is it better, when possible, to have two friends work together?
Is it better to have one slower and one faster student work together?
Should one poor and one better typist be paired?
Should partners alternate on activities or should each student work half of the period?
Which activities are necessary for both partners to do?
Should the entire class stay together on activities or should the partners be "let loose" to work at their own speed?
Where do these fit in: individual abilities, experience, skills, interest?

Each teacher felt s/he would be well enough acquainted with the students to answer the questions by the time classes began. As it turned out, each teacher handled the questions in different ways. And each teacher handled each situation for his/her individual classes in a different manner. Expediency dictated the classroom management systems. By later report, the chosen methods were effective for each class.

Prior to beginning the 3-week computer unit the students were administered a survey to determine their knowledge of and attitude toward computers. This was followed at the unit's end with a post-survey. When the computer classes began, the response from students, administrators, teachers, and parents was overwhelmingly favorable. Many problems and situations which had been anticipated never materialized. Questions of discipline, management, and coveraged seemed to disappear as the students in each class became engrossed with each day's assignment. The partnerships in system usage ran smoothly. In some classes there was an extra computer or two on which students who were ahead or behind could work individually. Students received help as needed from each other as well as the teacher, who circulated around the tables which had been placed in a U-shaped pattern. Students from the eighth grade PRIDE programs were also on hand in some classes and between periods to lend assistance.

Before school each day students were eagerly waiting outside the computer laboratory. The students were encouraged to write short programs using the day's concepts, and many did. One of the teachers collected these and shared them with other classes. Several parents, who were also teachers in PNISD and who were taking the night class, had children in the eighth grade. There was much discussion of the classwork in these homes as parents and students compared notes. Overall, the program was well-received, and all the students were reluctant for the unit to end.

Scheduling conflicts plus other considerations made it unfeasible to include the eighth grade science self-contained special education students with the other students. So the PRIDE teacher took them through a condensed version of the course during their science class.

After the course, when the computer survey was given again, a comparison of a sampling of opinions on the pre- and post-surveys indicated that in a few areas attitudes toward computers had changed slightly. Evidence of "computer phobia"

seemed to be lacking in both survey results. Students' awareness of computer advertising, computer impact on their daily lives, and the usefulness of knowledge about computers began and remained high. The biggest change was that a healthy doubt concerning human infallibility in computer usage increased.

The final round of classes was finished just before the Christmas holidays. A meeting was held after the holidays with the eighth grade science and PRIDE teachers to evaluate the program. Generally, the science teachers felt that the objectives which had been set were met and achieved by the course. Several suggestions were presented for consideration for change during the following year. The teachers were unanimous in the following: Scheduling of the unit should begin in the middle of a grading period and possibly be delayed until the second semester. They also asked that the lab be set up for one extra week on each campus to accommodate days off from school, days with shortened class periods, and student illnesses. They suggested that an individual book be issued and checked out to each student.

These were suggested by one or more of the teachers: They could use homework exercises and short quizzes on each chapter. These might or might not be used but would be available as the need arose. Also, they asked for more supplementary exercises for the appendix. Although some of the teachers liked students doubling on computers, they asked that more computers be allocated to the classes. Some felt that the optimum would be one system for each student. It would also be ideal (if the alarm system could be installed in the science labs) if the computer unit could be conducted there so two students could be located at each lab table. It was recommended that the possibility be explored of putting several of the longer chapter programs on the demonstration disks. In addition, a study should be made of the feasibility of having video tapes (either commercial or teacher-made) of the introductory lessons.

There was agreement that the booklet itself and demonstration disks were satisfactory as to format and inclusion. However, the booklet will be revised this summer both to correct some typographical mistakes and possibily to add explanations for various programs.

To date the program has proved to be a success given the scope and objectives originally envisioned. Interest among sixth and seventh grade students in keen, and they are eagerly looking forward to their opportunity to participate in the eighth grade earth science computer unit.

GUIDELINES FOR SUCCESS:

THE TRAINING OF FUTURE AND PRACTICING TEACHERS IN COMPUTER USAGE

by
Brad Chilton, Ed.D.
Assistant Professor of Computer Information Systems
Tarleton State University

ABSTRACT OF PRESENTATION: The focus of this presentation will be to identify the essential elements necessary to retrain today's practicing teachers and also to train new teachers so that they may successfully implement curriculum revisions involving computer instruction.

Recent approved legislation in Texas and recommendations for computer literacy in national reports causes the need to retrain present teaching staffs and also indicates that it is necessary to prepare relevant teacher education programs in computer literacy. In the past, computer literacy training has been primarily the responsibility of mathematics and business teachers, who have had little or no training in the concepts of computer usage.

The preparation of future teachers with courses relevant to their needs is a must. Many programs in the past were not oriented toward the training of classroom educational computer usage. The education of teachers in the future will have two primary areas of interest -- (1) computer literacy training of all teachers in all disciplines and (2) the professional preparation of computer literacy teachers. In regards to the computer literacy of teachers from other disciplines, it is imperative that their training be geared toward their individual discipline so that when they graduate, they may be able to implement computer literacy within their own discipline in the classroom. Past experiences have shown us that many educators are unable to relate their university computer literacy experience actual classroom usage. A program in the education of teachers

will be successful when teachers apply what they learn about the uses of computers in the classroom.

In order to prepare teachers from other disciplines, at least two college courses are necessary. One course should present a generalized introduction to computers and develop elementary programming concepts. Many present university introductory computer courses could meet these criteria The second course should relate the students discipline to computer usage. Relevant software and hardware should be studied. Students should be shown how to use computer equipment in the everyday activities of the educational setting.

The certification of teachers in the teaching of computers should be geared to the school setting. The constant development of new hardware and software programs must be constantly reevaluated. Currently, programs should present an overview of the operations of all sizes of computers but the focus of the program should be on microcomputers. Considering an eight-course certification program, the following courses are recommended:

(1) An introductory computer course. This course will give future teachers a broad perspective of the operation, terminology, and programming of different computer systems.

(2) Since microcomputers are widely used in the schools today, the language of microcomputers, BASIC, should receive full attention in a course of study.

(3) In order to further the prospective teachers understanding of computer languages, the student should receive training in a high level language such as Pascal, FORTRAN, or COBOL.

(4) A course in an assembler language should be included.

(5) Another course should be a study of the management of microcomputers and microcomputer systems.

(6) A study of computer assisted instruction would also be of great value since it would afford the future teacher with experience in the instructional uses of computers. Teachers

should be trained in the evaluation of software packages within the framework of this course.

(7) Two additional courses could be computer electives which would allow the student to pursue a particular area of greatest interest.

One of the most important facets of the program is its relevancy to the environment in the schools of today. In order to retain the relevancy features, the program should be constantly updated.

The preparation of present teachers should follow the guidelines listed above. Teachers from other disciplines need training as well as teachers who are currently certified in the teaching of computers. Colleges and universities can assist in this endeavor. Inservice programs can produce training activities and should also be utilized to provide "updates" as computer technology changes.

An additional problem that will be faced by school districts is funding. Hardware and software will require an expenditure of funds, and most school districts will require assistance from the state. Smaller school districts will need to form cooperative programs with other school districts in order to share facilities, hardware, and software.

In conclusion, the preparation of all teachers in computer literacy should be one of our most important objectives in education. As computer literacy is recognized as a "basic" of education, we must assure ourselves that our children are provided with the most up-to-date and comprehensive training that is available.

MICROCOMPUTER USAGE AND UTILIZATION IN NORTH CENTRAL TEXAS

by
Brad Chilton, Ed.D.
Assistant Professor of Computer Information Systems
Tarleton State University

ABSTRACT OF PRESENTATION: The focus of this presentation will be to report the results of a study conducted during the fall of 1983 regarding computer utilization in the public schools of north central Texas (50 public school districts surrounding and including Dallas and Fort Worth).

The age of the microcomputer has arrived in our educational system. National studies indicate the number of classroom microcomputers in our country is doubling every year. In order to determine the extent of the "microcomputer revolution" in north central Texas public schools, several studies have been conducted during the past two years.

A 1981 survey of the school districts in Dallas County showed that 87 percent of the school districts had computers available for student use. In the spring of 1982, a survey of the school districts included in Region XI indicated the presence of 400 microcomputers. A similar study six months later, indicated the presence of 794 microcomputers, an increase of almost 100 percent in the same region.

During the fall of 1983, a study was conducted in 48 school districts in Dallas, Tarrant and surrounding counties. Forty-five school districts responded to the questionnaire. Areas of study included the type of equipment utilized, courses offered, enrollments and trends, monetary expenditures and future plans.

Seventy-three per cent of the responding schools had microcomputers implemented for academics. Urban size school districts (classifications 4A or 5A) showed an implementation rate of 93 per cent. Of the 27 per cent of the

of the school districts who had not implemented microcomputers for instructional uses, most of the districts indicated plans for implementation within five years. The most popular microcomputer brand was Apple followed by TRS-80, Commodore and IBM. School districts reported phenomenal rates of growth, some as high as 600 per cent. Future expenditures indicate the rate of growth will continue to increase as many school districts reported future plans to budget in excess of $500,000.

In conclusion, computer technology appears to becoming a standard part of our educational setting. With expenditures expected to increase during the forseeable future, it seems that computers will be an "inescapable partner in education."

P U R C H A S I N G A M I C R O C O M P U T E R ?
--BE CAREFUL--DON'T GET AN EDSEL

by
Brad Chilton, Ed.D.
Assistant Professor of Computer Information Systems
Tarleton State University

ABSTRACT OF PRESENTATION: The focus of this presentation will be to present an independent view, free from manufacturer bias, on guidelines that should be followed in purchasing a microcomputer or microcomputer laboratory for instructional or administrative uses.

With the recent demise of the Texas Instruments' 99/4A microcomputer and the bankruptcy proceedings instigated by Osborne Computers, educators should consider new criteria in their purchase of microcomputers. Purchasers should carefully evaluate their investments to make sure they are making the best possible selection or they may end up with the 1980's model of the famous "Edsel".

Hardware considerations should be included in any decision regarding the purchase of microcomputers. Hardware involves the actual machine components--the items you can "touch" In general, it is best to have 64K of random access memory. If there is less, one may find ample memory will not be available for complex programs. Another consideration should be--Will the machine operate in a standard educational environment without requiring special air conditioning or unusual power supplies? The operating system is what controls the operation of the computer--from storing to processing information. If a communications system option is available, your computer will have the compatibility of communicating with other computers.

In order to store your information and make "backup" copies of your files, two floppy disk drives are recommended. A standard typewriter keyboard is recommended for ease of use. A separate

numeric keypad is helpful in business office applications and in mathematics classes. The video monitor should measure 24 lines with 80 characters on each line. A color monitor also enhances classroom educational activities In selecting a printer, one should determine the quality of output desired and the speed one needs for produced results. Letter quality printers will be expecially helpful in business applications while quiet nonimpact printers (such as thermal printers) might be appropriate for classroom use.

In choosing software (programs that run the machine), one should make sure the software is currently available and also that it operates efficiently on the microcomputer. Phrases from vendors such as "under development" or "to be released soon" should signal a warning since the development process may take years, if in fact, the software is ever developed.

In evaluating classroom software, be sure to test it yourself. A vendor "showcasing" software to you may show its "fancy" features but the salesman may fail to demonstrate its hidden "bugs". In examining educational classroom software, pretend you are the student and do such things as purposefully try to "confuse" the program. In business applications, attempt to determine if the program meets your needs for the current operating standards of your office. Remember software should not dictate how your office is run or the old phrase of the "tail wagging the dog" may apply. Software may be custom written for the job but with the large amount of manufacturer prepared software this is usually unnecessary.

Soliciting opinions of other educational users is an excellent avenue for review. The Texas Educational Computer Cooperative (TECC) has also developed a "checklist" for assisting in the development of software. In general, regarding software, test it yourself on your microcomputers to ascertain that it meets the criteria you have established.

Purchases from a large manufacturer, that has been in the business

for a number of years, does not guarantee excellent results. You must decide what type of service can be provided in your local area. If the vendors are financially unstable, eliminate them from consideration. If you have questions about a firm's financial status, check the company's earnings, recent sales, and obtain a financial statement. In regards to vendor consideration, note the following criteria:

availability of service contracts

type of servicing available

vendor's ability to modify software they sell

availability of training

installation assistance

guarantees

number of years in business

written contracts

list of other similar users

compatibility with larger computers

In conclusion, when purchasing a microcomputer, buy from someone you know and trust--a person who is interested in your educational setting. Be sure to purchase a microcomputer from a company with a good "track record" or you may be "stuck" with a 1980 model of the Edsel.

READIN', 'RITIN', AND 'RITHMETIC MOVE OVER FOR A NEW BASIC

by
Brad Chilton, Ed.D.
Assistant Professor of Computer Information Systems
Tarleton State University

ABSTRACT OF PRESENTATION: The focus of this presentation will be to report state and national developments that indicate that computer literacy is truly becoming a new basic of education.

Revolutionary changes are currently sweeping our educational system. At the forefront of these changes, is the current trend toward enhanced computer literacy training in the entire school curriculum.

The National Commission on Education Report recommends a comprehensive computer literacy training program. Computer literacy as a "basic" of education along with the standard required courses is included in this report. For high school graduation, a one-half year course in computer literacy is recommended. H. Ross Perot's Select Committee on Public Education and House Bill 246 also are giving an indication of the of the emergence of computer literacy as a new basic of education. The traditional basics as well as computer literacy are included in the educational setting for all years of the students study. A review of House Bill 246 discloses the presence of computer literacy training from kindergarten through high school. A computer literacy course will be required for graduation from high school.

The former superintendent of the Dallas Independent School District, Nolan Estes, says that computer literacy is becoming a fourth basic of education. Another indication of the computers' importance to our life is instead of selecting a man of the year for 1982,

Time magazine selected a machine of the year--the computer. A recent Yankelovich Poll indicates that nearly 80 per cent of Americans expect in the fairly near future that home computers will be as common place as television sets or dishwashers. The computers' importance to the worldwide education system is indicated by French editor Jean-Jacques Servan-Schreiber who believes the computer's teaching capability can conquer the Third World's illiteracy problems.

In conclusion, the impact of computer technology will cause the educational system to undergo more changes in the next few years than have occurred in the past half century. If these changes are made in an efficient manner, the educational opportunities afforded children will be enhanced.

A TIME OF CHANGE:

COMPUTERS CHALLENGE TODAY'S EDUCATIONAL ADMINISTRATORS

by
Brad Chilton, Ed.D.
Assistant Professor of Computer Information Systems
Tarleton State University

ABSTRACT OF PRESENTATION: The focus of this presentation will be to outline various guidelines for educational administrators and teachers to utilize in implementing curriculums featuring the usage of computers as relating to the current revisions in the state guidelines for curriculum instruction in Texas.

Educational administrators are currently faced with many problems concerning computer literacy training. The needs for competent teachers and adequate hardware and software pose significant obstacles in implementing a successful program of study.

In starting a new program of study, the first priority should be to meet state standards and objectives. Current state standards and measures which are likely to be implemented advocate programs which will include all grade levels, kindergarten through high school.

Members from all areas of your teaching staff should be included in your planning process. For instance, hardware for kindergarten students will be different from that required for high school students. Kindergarten students would probably function best with a membrane style keyboard while high school students would require a typewriter style keyboard. Input from staff members will help to insure the best possible implementation process.

The appointment of a district computer literacy and curriculum committee should be an initial step in the planning process. Composed of teachers and administrators, this group will evaluate decisions relating to computer curriculums and equipment. This will insure that the districts computer program is a coordinated program. In larger dis-

tricts, a full-time district coordinator will be able to oversee the computer program. This person should serve on the district computer service committee, and should be responsible for implementing the computer curriculum. In addition, the coordinator will support teachers in acquiring and utilizing materials. This person will also provide suggestions on instructional methodologies. Additionally, the committee and coordinator will study computer utilization and make recommendations regarding their use.

The use of short and long range objectives is a popular method of organizing the implementation of a program. Long-range objectives should be broad and should mainly pertain to student competencies as equipment procurement will be difficult to define in long range objectives because of the constantly changing technology. Short-range objectives can be very specific in regards to student learning objectives, hardware and software.

Many small school districts will face staff and funding problems. It may be that cooperative programs among districts will provide one solution. With the portability of microcomputers, the microcomputer laboratories may be transported from school to school or district to district on a weekly or even a daily basis. On a lesser scale, a number of large districts will face similar problems as do the small school districts.

Educational administrators must develop organized methods in order to implement computer literacy programs. Programs of the future will be more encompassing than present programs, the administration of these programs will demand a great amount of time, which will require full time administrative support in most school districts.

Microcomputer Literacy for the Educational Administrator: Applications for Microcomputer Systems for Noninstructional Tasks

K. Lyle Froese, Ed.D.
Sherman Independent School District
P.O. Box 1156
Sherman, Texas 75090

Although microcomputers have entered public schools as an important process, their use for increasing administrative efficiency and effectiveness has been limited. Factors which may contribute to this situation are: (1) lack of a convenient information base for selection of administrative microcomputer software, (2) hesitance of administrators to become involved in a rapidly changing technology, and (3) inadequate staff development for administrators on microcomputer utilization.

The introduction of the relatively inexpensive personal computer, or microcomputer, in the late 1970s marked the beginning of increased computer utilization for a large number of public school users. The National Center for Education Statistics of the U.S. Department of Education reported that approximately one half of the nation's school districts provided at least one microcomputer or computer terminal for their schools (Goor, 1982:20). Data from the study indicated that 87 percent of those school districts utilized microcomputers and that 39 percent utilized only the more traditional computer terminals.

Although it has been shown that microcomputers have been placed in schools with increasing frequency in the past years, the purpose for these microcomputers primarily has been instructional. A study conducted by the Center for Educational Management Research and Training (Corbett, 1982:111) revealed that of the approximately 580 microcomputers in use in schools surveyed, only four school administrators had microcomputers in their offices and were making use of them. Some schools visited by the Center's staff members had as many as twelve to fifteen microcomputers in the classrooms and none at all the the office for word processing, scheduling, or attendance accounting. A need for increased utilization of microcomputers by Texas educational administrators was reported by Froese (1983:98) in a survey of members of the Texas Education Computer Cooperative Advisory Committee.

Poirot (1981:31) stated that computer technology is effective for administrative tasks because of the large amounts of information to be processed and the kinds of repetitive operations needed. Together with traditional business, mathematical, and statistical computer applications, software systems can accomplish tasks of information control, reporting, operations research, feasibility simulations, modeling, and generally more-efficient decision making.

The question of what school administrators should learn about computers to become computer literate has been debated since computers were first introduced into the schools. Most educational technologists agree that computer literacy should include:

1. Becoming familiar with the administrative tasks the microcomputer can be expected to perform; and
2. Becoming experienced in using the microcomputer with packaged administrative programs.

Computer programming skills are not generally needed in order to be identified as an educational administrator who is computer literate. A differentiated computer literacy program may include programming, but only some staff members should be expected to develop microcomputer programs for specific purposes. Hopefully, the fear of being required to possess computer programming skills in order to be computer literate will be diminished through acceptance of skills that all should possess and skills that some should possess.

Staff development for all administrators should be provided by school districts and regional centers. A series of five inservice sessions for educational administrators may be developed around the following objectives:

1. Helping educators feel comfortable using a microcomputer.
2. Familiarizing educators with good quality administrative software.
3. Teaching educators how to set up and operate a microcomputer system.
4. Introducing educators to general application programs for customizing.
5. Teaching educators how to duplicate authorized programs and data.

A few years ago, software for educational administration was limited in quantity and quality. Specific application software is being produced today by a growing number of programmers. The overall quality of most programs is improving somewhat, but administrators should be analytical in their acquisitions. The largest increase in software has been in the use of general application programs for a variety of uses. Included in these general applications programs are: (1) data base management, (2) spreadsheet calculation, and (3) word processing.

The advantages of general application programs for educational administration were identified by Pogrow (1982:56). Since there is a larger market for general programs as compared to specific use programs, general programs tend to be more widely available and generally of higher quality than specific programs. General application programs are more cost-effective since the same program can be modified to handle many different tasks. As an administrator becomes more proficient at using general application programs, increased capabilities of the system often result. General application programs are not dependent upon one programmer's perspective of a task, as is often the case with specific application programs. Individual needs can often be met by the customizing feature of general programs.

Data base management programs could be utilized for a variety of administrative tasks. School offices characteristically maintain a large number of records and files. The advantages of data base management programs include easy access to information, updating of information, searching, sorting and reporting information. Some of the typical administrative tasks include:

1. Personnel records
2. Student records
3. Substitute teachers
4. Discipline records
5. Equipment inventory
6. Resources file
7. Vendor file
8. Applicant file
9. Calendar of events

The selection of a data base management program should be determined by the characteristics of the administrative task. Some of the questions to be asked before selecting a data base management program include:

1. How much information will be included in each record?

2. How many records will be stored in the file?

3. How will the records be sorted and reported?

4. Will common information on different records be substantial?

5. Will information be updated, expanded, or deleted often?

Another general application program which can be customized to meet individual needs is the spreadsheet. Rows and columns of data may be operated on by mathematical formulae to produce totals, percents, and increments. The rapid calculations make possible a wide range of optional or trial runs on the data. Some of the possible uses of spreadsheet calculation programs for educational administration include:

1. Budget planning
2. Test score analysis
3. Attendance analysis
4. Cash management
5. Departmental expenditures
6. Class load projections
7. Energy cost analysis
8. Salary planning
9. Gain score per cost analysis

Some of the questions to be asked before selecting a spreadsheet program might include:

1. How many rows and columns of data will be entered?

2. What mathematical operations will be performed on this data?

3. What reports will result from these calculations?

4. Will the data be transfered from, or to, another program or file?

5. How much information would be needed on one screen at a time?

The third type of general application program is word processing. Text information that is often edited or repeatedly produced in some modified form could be handled very efficiently by a word processing program. Some microcomputers are better suited for word processing than others. The number of characters per line of text is important in text editing. Upper and lower case print format is required for most text. Some programs allow lists of information from other sources to be included within text material. Some of the administrative applications of word processing programs include:

1. Curriculum guides
2. Student/parent handbooks
3. Correspondence
4. Supervisory reports
5. Staffing narratives
6. Publications
7. Forms control
8. Customized mailings
9. Job descriptions

Most word processing programs provide for text editing by character, word, paragraph, or block. Some programs include underlining, subscripts, superscripts, bold type, and spelling checking. Consideration should be given to the type of printer to be utilized. Some of the questions to be asked before selecting a word processing program include:

1. What maximum width should be shown on the display?

2. Will upper, lower, and special case characters be necessary?

3. Will information from other programs and files be needed?

4. Will printing speed or printing quality be important?

5. How much information will be stored in each text file?

Programs specifically designed for educational administration were identified by Froese (1983:68) through letters of inquiry to microcomputer software suppliers. Specific application programs were identified in the areas of financial accounting, attendance recordkeeping, grades management, inventory, planning/projecting, student scheduling, and student records. Each of the programs was designed to be utilized with one or more specific microcomputer systems. Most of the programs were developed for the Apple II microcomputer. The next most frequently found software was designed for the Radio Shack microcomputer. Costs for specific application software

varied from $29.95 to $3,995.00. Some software packages included several programs. Higher cost software often included on-site orientation and consultation for a period of time.

The specific administrative task which was performed by the largest number of programs was student attendance recordkeeping. The least frequently identified specific application was for inventory management. Programs designed for student scheduling often required additional capabilities of the microcomputer system, such as larger memory, larger number of diskette drives, special operating systems, and card readers. Costs for scheduling programs were significantly higher than other specific application programs.

At a time when education is much discussed for either the existence or lack of merit, administrators must not allow the potentials of technology applications to be lost in the debate. Computer literate administrators can go forward and utilize the benefits of technology to make the significant differences expected of them. The application of microcomputers to administration can increase task effectiveness and efficiency--and enjoyment.

References

Corbett, John J., and others. May, 1982. "Principals Are Key to Effective Use of Computer Technology." NASSP Bulletin, 66:109-115.

Froese, K. Lyle. 1983. "Microcomputer Literacy for the Educational Administrator: The Applications for Microcomputer Systems for Selected Noninstructional Tasks." Doctoral Dissertation, East Texas State University.

Goor, Jeanette, Arthur Melmed, and Elizabeth Farris. 1982. Student Use of Computers in Schools. Washington: U.S. Department of Education.

Poirot, James L. 1981. Computers and Education. Manchaca, Texas: Sterling Swift.

Pogrow, Stanley. September, 1982. "Microcomputerizing Your Paperwork: Easy, Economical, and Effective." Electronic Learning, 2:55-59.

COMPUTER-ASSISTED INSTRUCTION (CAI) STUDY SKILLS WITH HIGH SCHOOL STUDENTS

Bernadette M. Gadzella, Ph.D.
East Texas State University
Commerce, Texas 75428

This presentation is a report on the findings of two studies done with computer-assisted instruction (CAI) study skills modules with high school students. A total of 104 students participated in the studies. Results showed that students gained significant insights from the instructions in study skills presented through CAI as measured by a test.

Students who succeed in school know how to study. Many students fail not because they lack intellectual ability, but because they do not know how to organize, assimilate and retain the information they attempt to learn. In short, many students fail due to their poor study habits.

Although students can learn study skills by themselves, it is often a long, frustrating experience of trial and error. One does not naturally acquire effective study skills, such skills must be taught. Once effective study skills are acquired, students can proceed independently in learning and can profit more from instructions when they are provided.

Over the years, studies (Brown, 1974; Gadzella, 1979; Gadzella, Goldston & Zimmerman, 1977; Zimmerman, Goldston & Gadzella, 1977) have been conducted on the effects of study skills instructions on college students' attitudes and behaviors. Many approaches were used to disseminate the information on study skills: class instruction, and group and individual counseling. Recently, the use of the computer in disseminating the information on study skills has shown to be much more effective and efficient

[1] Studies were supported by ETSU Organized Research Grants Nos. 1501-9206 and 1501-9303.

(Gadzella, 1980; Gadzella, 1981; Gadzella, 1982).

The purpose of this presentation is to report the findings of two studies in which computer-assisted instruction (CAI) on study skills were used with high school students.

The CAI Study Skills Modules (Gadzella, Brown & Cochran, 1981) used in these studies, were developed with some basic principles of learning in mind. It is assumed that learned occurs more readily when (a) students participate in the learning process, (b) students feel a sense of satisfaction, (c) students are aware of their progress in what they learn, (d) materials are presented in meaningful ways, and (e) materials are presented in a pleasant and enjoyable fashion.

Ten topics related to basic study skills are presented in the CAI modules: time management, memory improvement, note taking, textbook reading, examination taking, report writing, oral reporting, scholastic motivation, interpersonal relations, and concentration improvement. In the CAI modules, the computer acts as a transmitter of knowledge about effective study skills. Each module ends with a quiz of ten multiple-choice items. The computer scores the quiz and interpretes the scores for the students.

As the students do through the CAI modules, they enter the information on effective study skills and how to develop such skills in a CAI Study Skills Notebook (Brown, Gadzella, Forristall & Cochran, 1981). Thus, students are actively participating in the program and will have the information for future refrences.

A CAI Study Skills Test (Brown & Gadzella, 1981) was developed to measure each of the ten topics presented in the CAI modules. Thus, the test consists of ten scales scores and a total test score. The test can be done on specific answer sheets or on the computer.

Study 1

Fifty-six students enrolled in two advanced senior high school English classes in a large high school in Texas agreed to participate in this

study. Both classes were taught by the same teacher. Students were randomly assigned to experimental and control groups. Of this group, 50 students completed the study, of which 23 were in the experimental group and 27 in the control group.

All students took the study skills test prior to the study. The experimental group completed the CAI modules and notebook. Then, all students took the study skills posttest. This was completed in four months and these data were analyzed even though the subjects in the control group were given a chance to do the CAI modules.

Data for the pre- and posttests for the study skills scales and total test were analyzed using the analysis of covariance. Results showed that the experimental group scored significantly higher on the total test and on the Memory Improvement and Interpersonal Relations scales.

Study 2

The subjects were 57 students enrolled in two computer math classes and one algebra class in a high school in Texas. The three classes were taught by one teacher. Students were randomly assigned to experimental and control groups. Of this group, 54 students completed the study. In this group of 54, there were 28 males and 26 females. In this group, 11 were in Grade IX, 5 in Grade X, 18 in Grade XI and 20 in Grade XII. Their average GPA was 2.80. Of this group, 28 students had a GPA of 2.80 or above and 26 had a GPA of 2.79 or below.

In this study, all students took the study skills test and the Tennessee Self-concept Scale, TSCS (Fitts, 1964) before and after the experimental group completed the CAI modules. After the experimental group completed the modules, the control group did the CAI modules. After the control group completed the modules, all students responded to the study skills test and TSCS. GPA were provided by the school administrative office. The data were analyzed using the *t*-test and the product-moment correlation.

The results of the analyses showed that the total group gained significantly from doing the CAI Study Skills Modules and indicated a greater self-satisfaction (on the TSCS) with themselves after they completed the CAI program. For various groups, the results showed that (a) the experimental group gained significantly more on study skills when compared to the control group, (b) Seniors gained significantly more than juniors, and (c) high achievers (GPA of 2.80 and above) gained significantly more than low achievers (GPA of 2.79 and below).

GPAs were significantly related to the total study skills test scores and to eight study skills scales on the posttests. GPA did not correlated significantly with the total self-concept score but did correlate with the Moral-Ethical subscale of the TSCS.

Summary

Several observations can be made from these findings and those reported in previous studies. Effective study skills are related to academic achievement for high school students and for college students. High, positive self-concepts do not cause high academic achievement, but it is a vital quality to have prior to achievement. Teachers might familarize themselves with the general area of self-concept dynamics and how it is related to behavior. In addition to providing information, teachers might also assist students by providing inspiration and direction and offer some hope of success. Success in school leads to self-satisfaction and effective study skills help students to achieve academic success.

Other studies with larger number of students should be conducted. However the data in the studies reported here show that the computer can effectively present the information on study skills to high school students. Program designers might take that into consideration and include such a program in their high schools.

REFERENCES

Brown, W. F. Effectiveness of paraprofessionals. The evidence. Personnel and Guidance Journal, 1974, 53, 257-263.

Brown, W. F., & Gadzella, B. M. CAI study skills test. San Marcos, Tx: Effective Study Materials, 1981.

Brown, W. F., Gadzella, B. M., Forristall, D. Z., & Cochran, S. W. CAI study skills notebook. San Marcos, TX: Effective Study Materials, 1981.

Fitts, W. H. Tennessee self-concept scale. Nashville, TX: Counselor Recordings and Tests, 1964.

Gadzella, B. M. A CAI study skills program. ERIC ED 211 059, 1981.

Gadzella, B. M. A comparison of CAI and class instruction approaches on study skills. ERIC Clearinghouse on Information Resources, 1980.

Gadzella, B. M. Computer-assisted instruction on study skills. Journal of Experimental Education, 1982, 50, 122-126.

Gadzella, B. M. The effects of student-to-student counseling on student perceptions of study habits and attitudes. Journal of College Student Personnel, 1979, 20, 424-430.

Gadzella, B. M. The effects of study habits and attitudes instruction on university students and their academic performance. ETSU Organized Research Report, No. 1501-9240, 1977.

Gadzella, B. M., Brown, W. F., & Cochran, W. S. CAI study Skills modules. San Marcos, TX: Effective Study Materials, 1981.

Gadzella, B. M., Goldston, J. T., & Zimmerman, M. L. Effectiveness of exposure of study techniques on college students' perceptions. The Journal of Educational Research, 1977, 71, 26-30.

Zimmerman, M., Goldston, J., & Gadzella, B. M. Prediction of academic performance for college students by sex and race. Psychological Reports, 1977, 41, 1183-1186.

The Fifth Generation and Expert Systems

by
Dr. Philip Gehring
308 Westwood Terrace
Austin, Texas 78746

ABSTRACT

This is a synthesised document of several recent scholarly papers about fifth generation computers, expert systems and the potential impact of new technologies on education and society in general. These are not papers which would generally be read by educators. However, it is important that educators begin to comprehend that the present infusion of technology into education is plebian compared to what the future will require that we be able to advantageously implement.

INTRODUCTION

This paper is cut and pasted from several excellent articles I have read regarding "Fifth Generation" computers and "Expert Systems". The original authors are listed in the references. Some personal observations and rewrite have also been included. The reason for preparing this synthesised document is to bring to the attention of educators some awareness of the dramatic changes in store in computer and information technology. We have all begun to realize how pervasive and helpful computers can be across all academic disciplines from elementary school through the university. The thrust of the research in hardware architecture, software and artificial intelligence should give us pause to consider the unlimited potential for application to education and learning--and the change in education that will likely accompany these innovations. Unlike other so called revolutions in education we will not be permitted an indifferent implementation of these technologies. Therefore, we must know what is coming and work diligently to insure it is applied to learning in a productive way. Our hopes and expectations should always exceed our fear of change.

EDUCATION AND THE ELECTRONIC AGE

> "What is there in the list of strange and unexpected events that has not occurred in our time? Our lives have transcended the limits of humanity; we are born to serve as the theme of incredible tales to posterity."
> - Aeschines (389-314 B.C.)

The following is a summary of a scholarly article by Harold Shane. His observations form an excellent preamble to the subsequent information about technological change which supports his theme.

The Harvard sociologist, Daniel Bell, coined the term "post-industrial society" to describe an era based on services, into which our industrial age is being transformed. However, it has become clear that the post-industrial world is not just a service society; it is an "information society" -- one in which knowledge and the handling of information have elbowed aside the smokestack and the assembly line as symbols of America's prowess.

The unique opportunities (and demands) that accompany the information era require us to explore and to understand more clearly the myriad ways in which society is being influenced by the information sciences and microtechnologies. We must determine what social inventiveness and institutional repair work may be required because of 1) the enormous impact of the microprocessor and of telecommunications networks, 2) ergonomic responses to increasingly robotized workplaces and computerized homes, and 3) ubiquitous 24-hour television, which, in some areas, now allows access to more than 100 channels.

Another challenge in an era of information overload is meeting the demands that will be placed on education. As Bentley Glass put it in 1968, "The educated man of yesterday is the maladjusted, uneducated man of today and the culturally illiterate misfit of tomorrow." Consequently, educated humans need to acquire much more knowledge than they needed to have 25 or 30 years ago. Traditional curriculum content and instructional practices are certain to change as educators begin to master the art of using technology to react promptly and wisely to the difficulties created by the demands that accompany an era of information overload.

Health and longevity are also being influenced in the microelectronic epoch. Computerized prosthetic devices have already helped the paraplegic to walk, the voiceless to speak, and the brain-injured to function. These applications as well will have a profound impact on the education of the handicapped and other special needs groups.

The long-term impact of rapid technological change is nearly impossible to forecast accurately. The technological changes reported almost daily tend to threaten us with technogenic unemployment even as they open new employment vistas. In July, 1983 for instance, Robotics World announced that scientists at the Nippon Telegraph and Telephone Public Corporation had created a robot that could read the news with 99.5% accuracy. The robot's computer (programmed with a dictionary) can scan print, check its memory banks, and deliver a properly inflected sentence. The report suggests a future for news-reading robots on the radio and for robots as telephone operators and as readers for the blind. Some innovations are already creating technogenic unemployment, as when, for instance, a factory is retooled and automated. Educators from the preschool to the graduate school will need to be increasingly alert to shifting demands posed by a rapidly changing workplace. As Alvin Toffler sees it, educators need to begin revamping the present system of mass education:

> "Today's schools are turning out still more factory-style workers for jobs that won't exist. Diversify. Individualize. Decentralize. Smaller, more local schools. More education in the home. More parental involvement. More creativity, less rote (it's the rote jobs that are disappearing the fastest)."

One other development of the silicon age (now gallium arsenide) that could soon affect all our lives is worthy of mention. Kenneth Hanck and Keith DeArmond, chemistry professors at North Carolina State University at Raleigh,

molecules. Such molecules would be capable of storing readable, coded information in approximately one-millionth of the space that even today's tiny computer chips require. Computers using molecular memory devices would be incredibly more powerful, tinier, and more efficient than anything now available.

In recent months our entire system of education - kindergarten through college - has come under harsh scrutiny. In part, this is because governments, institutions, and individuals are discovering that dynamic developments in electronic information technologies, in biotechnologies, and in complex global communications are rapidly changing what they do, how they do it, and how they relate to one another. But the onrushing age of the micro chip is already changing the entire landscape of education. According to the Office of Technology Assessment:

> "The so-called information revolution, driven by rapid advances in communication and computer technology, is profoundly affecting American education. It is changing the nature of what needs to be learned, who needs to learn it, who will provide it, and how it will be provided and paid for."

We cannot improve our schools without first determining how to cope with the computer and information age; we need to develop strategies - both as professional educators and as private citizens - for living and learning in an electronic epoch, as well as future-oriented administrative policies, curriculum content, instructional practices, knowledge sources, links with parents and community and other educational agencies - including the ubiquitous TV set and its video-appendages.

Basic skills, clearly defined, are of even greater importance in the new knowledge society that we are entering. As we define "basic skills," we find that, in addition to the basic computer skills necessary to an information era, familiar basic elements - e.g., reading and writing, which have constituted traditional literacy - may acquire new meanings.

The societal tensions generated by microtechnologies and their impact on our world will not grow less. Schools must begin living in the future now, just as their young charges will have to do. Even amid rapid and often bewildering changes, let us remain optimistic. We must debate and begin to determine what the "mircokids" of today and tomorrow need to know and how U.S. schools can best educate the "microelectronic generation."

The remaining summaries are taken directly from articles by various referenced experts appearing primarily in the IEEE Spectrum.

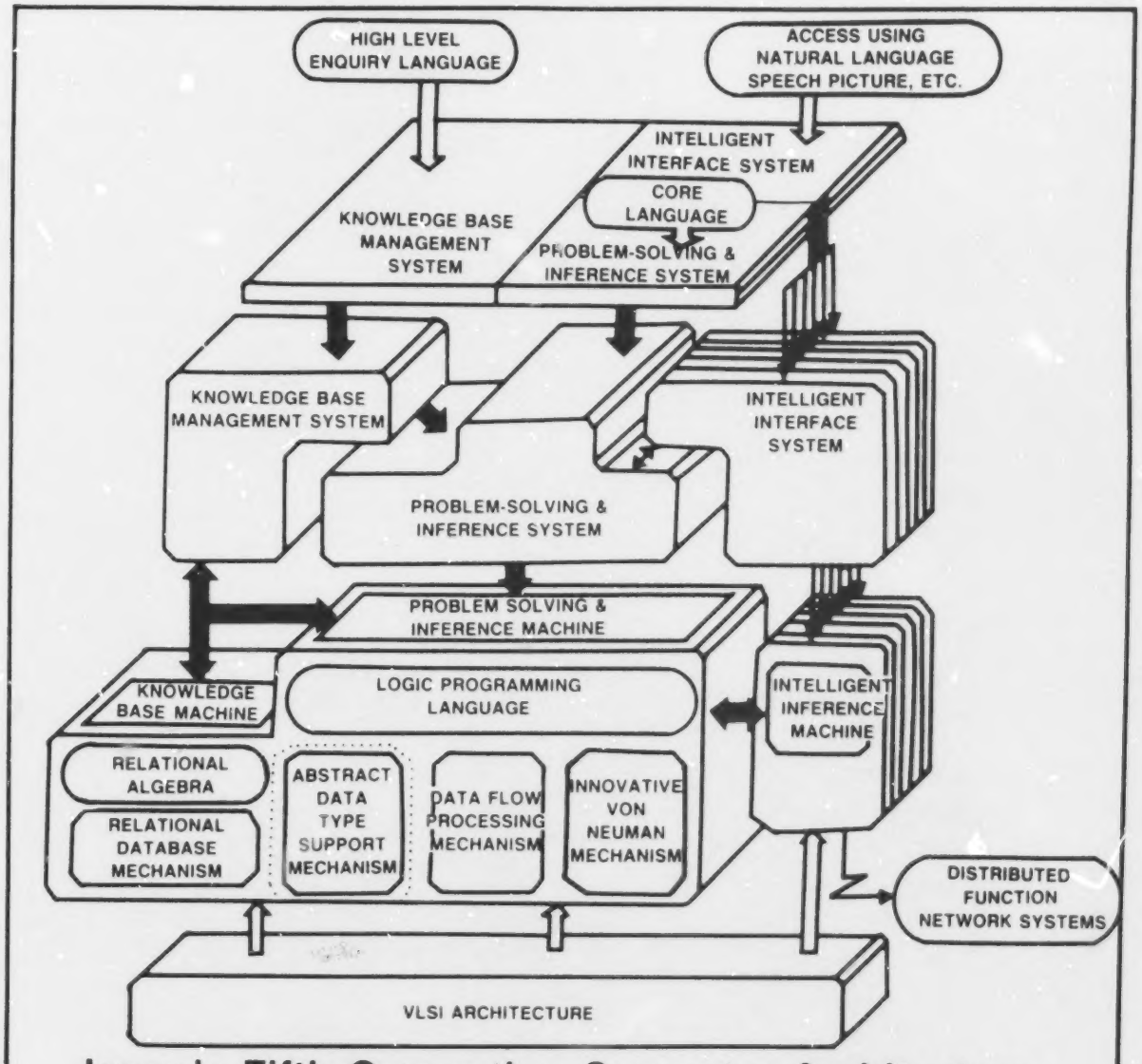
THE FIFTH GENERATION: A QUANTUM JUMP IN FRIENDLINESS

Computers are no longer used only for science and technology, business computations, and the automated control of industrial processes; they now have penetrated our daily lives and are becoming society's central nervous system. Machines are being developed that will interact naturally with human beings. Around the world the basic tools needed to fulfill this grand expectation are gradually developing. We anticpate that in Japan the new Fifth-Generation Computer Systems Project will stimulate rapid progress toward computer systems suited to the new era.

Fifth-generation computers are expected to be used predominantly in the society of the 1990s. In addition to being capable of processing numeric data, they will handle such non-numeric data as symbols, words, pictures, and human

speech. They will manipulate not only raw data, but also knowledge - that is, information judged or organized according to intrinsic characteristics, much as humans organize knowledge in their brains. Applied artificial intelligence will play a key role.

Fifth-generation computers are more than a direct extension of the computer technology that packs more integrated circuits into a smaller space. They are to be distinguished from the large and fast number-crunching supercomputers being developed for intricate scientific and engineering calculations and simulations.



Japan's Fifth Generation Computer Architecture

Fifth-generation computer systems are expected to fulfill four major roles: (1) enhancement of productivity in low-productivity areas, such as nonstandardized operations in smaller industries; (2) conservation of national resources and energy through optimal energy conversion; (3) establishment of medical, educational, and other kinds of support systems for solving complex social problems, and (4) fostering of international cooperation through the machine translation of languages.

The fields in which computers will be applied in the 1990s will be extremely diversified, ranging from individual components in appliances to giant information networks with worldwide connections. Everyone will be using computers in daily life without giving it a second thought. For that to happen, however, an environment will have to be created in which a human and a computer can communicate easily through a wide variety of information media such as speech, text, and graphics- in short, the common methods by which people communicate with each other.

The changes from one generation to the next in computer technology have so far been made to accommodate changes in various devices- that is, from vacuum tubes (the first generation) to transistors, then to integrated circuits, and recently to large-scale integrated circuits. Although computers have become faster and more complex, there have been no major changes in the fundamental design philosophy and objectives of computers.

The design philosphy behind conventional Von Neumann computers in use today called for a minimum of hardware to configure systems of maximum simplicity, capable of efficcient processing with software. New directions in computer technology have led to a rethinking of this design philosophy. First, the speeds of computers now used are approaching the limit imposed by the speed of light. Second, the emergence of VLSI technology has substantially reduced hardware costs, and it soon will be cost-effective to use as much hardware as is required. To take full advantage of the mass production of VLSI chips while computing faster than the limit imposed by the speed of light, parallel processing must be used instead of sequential processing. Finally, current computers are extremely weak in basic functions for processing speech, text, graphics, and other nonnumeric data; they are also weak on artificially intelligent processing, such as inference, association, and learning. It is believed that for computers to thrive in the 1990s, they must evolve from machines dedicated to numerical computations to machines that can assess the meaning of information and can understand the problems to be solved.

TOMORROW'S COMPUTERS

The unusually malleable set of technologies entailed in next-generation computing suggests that the ubiquitous computer- perhaps best symbolized today by the pervasive personal computer- seems destined to rise to a new level of influence and importance.

Cost-effective increases in machine speed, largely due to advances in the microelectronics field, have been widely visible, particularly with the advent of high-speed microcomputer systems. A major international industry in semiconductors has emerged, and supermacy in this field is often used as a barometer for leadership in high technology. Less noticeable perhaps, but not insignificant, has been the steady progress in the field of artificial intelligence (AI) over the last 15 to 20 years. This progress has given rise to a discipline, based mainly in the universities, that has been experimentally developing and demonstrating techniques for machine intelligence. We are now at a stage where the confluence of these two disciplines-microelectronics and

artificial intelligence- may indeed produce new generations of computers that are both fast and smart.

The nation that dominates this information-processing field will possess the keys to world leadership in the twenty-first century. In its brief history, information processing has affected virtually every facet of modern society. Yet we see almost unlimited opportunity for application of the technology. Better management and use of information is at the core of opportunity in the space environment. Improved health care, education, and training may also result from judicious use of advanced information processing.

We know now that machines can be built that can accept English or other languages as input and process them in near real time. This should greatly ease the problem of entering complex expressions into the computer. Machines that can see and understand their environment will have significant benefit in robotic applications that are far too numerous to mention. Advanced techniques for planning and reasoning are most likely to be developed and improved over time through actual application to real problems.

The three principal disciplines that will make a new generation of machines a reality are: (1) microelectronics, which will produce the fast, ultrasmall devices made up of large-scale chips and wafers; (2) artificial intelligence, which will guide the development and exploration of concepts and technniques for intelligent systems; and (3) computer systems and architecture that will guide development of the hardware, software, and related tools to realize working systems.

There is also that branch of artificial intelligence which relates to knowledge base sytems and expert systems. A knowledge base system, unlike a data base system, is a system which processes inferences. One can use as input many inferences and the machine will be able to survey these inferences and come to conclusions. One can then ask all sorts of questions and the machine will produce answers to the questions in the same way that a human expert, like a doctor, would carry out a medical diagnosis. Today we can see about a hundred examples of expert systems in use in different fields. The Japanese believe that in the 1990s, knowledge base and expert systems will constitute the majority of computer usage. What will intelligent systems look like? Unlike the monolithic supercomputers of today, these systems will be small, compact units that carry out a specialized function. The new generation of machines will consist of collections of these systems integrated into a whole. If a system needs vision, it can be added. Speech input can also be incorporated, but the speech system (and probably every component) will have to be given enough knowledge to do its job.

EXPERT SYSTEMS

In most areas of artificial intelligence- that branch of computing science that attempts to have machines emulate intelligent behavior- programs fall far short of the competence of humans or even animals. Computer systems designed to see images, hear sounds, and recognize speech are still in fairly early stages of development. However, in one area of AI- that of reasoning from knowledge in a limited domain- computer programs can not only approach human performance, but in some cases they can exceed it.

These programs use a collection of facts, rules of thumb, and other knowledge about a given field, coupled with methods of applying those rules, to make inferences. They solve problems in such specialized fields as medical diagnosis, mineral exploration, and oil-well log interpretation. They differ

substantially from conventional computer programs because their tasks have no algorithmic solutions and because often they must make conclusions based on incomplete or uncertain information.

In building such expert systems, researchers have found that amassing a large amount of knowledge, rather than sophisticated reasoning techniques, is responsible for most of the power of the system. Such high-performance expert systems, previously limited to academic research projects, are beginning to enter the software marketplace. AI programs are turning up for potential use in robots, medical diagnosis, an automatic crew-scheduling system for the U.S. Space Shuttle, and flight systems for F-16 military aircraft, among other places.

The structure of an expert system is modular: facts and other knowledge about a particular domain can be separated from the inference procedure- or control structure- for applying those facts, while another part of the system- the global data base- is the model of the "world" associated with a specific problem, its status, and its history.

It is desirable, though not yet common, to have a natural-language interface to facilitate the use of the system both during development and in the field. In some sophisticated systems, an explanation module is also included, allowing the user to challenge the system's conclusions and to examine the underlying reasoning process that led to them.

An expert system differs from more conventional computer programs in several important respects. In a conventional computer program, knowledge pertinent to the problem and methods for using this knowledge are intertwined, so it is difficult to change the program. In an expert system there is usually a clear separation of general knowledge about the problem (the knowledge base) from information about the current problem (the input data) and methods (the inference engine) for applying the general knowledge to the problem. With this separation the program can be changed by simple modification of the knowledge base. This is particularly true of rule-based systems, where the system can be changed by the simple addition or subtraction of rules in the knowledge base.

Rule based systems work by applying rules, noting the result, and applying new rules based on the changed situation. They can also work by directed logical inference, either starting with the initial evidence in a situation and working toward a solution, or starting with hypotheses about possible solutions and working backward to find existing evidence- or a deduction from existing evidence- that supports a particular hypothesis.

Not all expert systems are rule-based. Rule-based systems are particularly attractive when much of the expert knowledge in the field comes from empirical associations acquired as a result of experience. Where more causal information is available, then other representations may be more pertinent, such as networks that link state nodes (representing parts of the physical system in question) by arcs representing causal relations. Such networks allow systems to operate on a model of the situation in question.

One of the earliest and most often applied expert systems is Dendral. It was devised in the late 1960s by Edward A. Feignbaum and Joshua Lederberg at Stanford University to generate plausible structural representations of organic molecules from mass spectrogram data. Dendral has been used as a consultant by organic chemists for more than 15 years and has been cited in numerous articles. It is currently recognized as an expert in mass-spectral analysis.

One of the best-known expert systems is Mycin, designed by Edward Shortliffe at Stanford University in the mid-1970s. It is an interactive system that diagnoses bacterial infections and recommends antibiotic therapy. Mycin represents expert judgmental reasoning as condition-conclusions rules, linking

patient data to infection hypotheses, and at the same time provides the expert's "certainty" estimate for each rule. It chains backwards from hypothesized diagnoses, using rules to estimate the certainty factors of conclusions based on the certainty factors of their antecedents, to see if the evidence supports a diagnosis. If there is not enough information to narrow the hypotheses, it asks the physician for additional data, exhaustively evaluating all hypotheses. When it has finished, Mycin matches treatments to all diagnoses that have high certainty values.

All the approaches used to guide reasoning by expert systems are intended to reduce the amount of time spent looking for solutions among all the potential answers to a given problem; exhaustive computational searches are not appropriate because, even with a small number of different factors to consider, possibilities expand rapidly- the so-called combinatorial explosion.

Many expert systems are also under development. Their application areas include chemical and biological synthesis, mineral and oil exploration, planning and scheduling, space defense, air-traffic control, circuit analysis, VLSI design, equipment fault diagnosis, computer-configuration selection, speech understanding, and computer-aided instruction.

There appear to be few constraints on the ultimate use of expert systems. However, the nature of their design and construction is changing. The limitations of rule-based systems are becoming apparent: Not all knowledge can be structed as empirical associations.

The growth of expert systems, coupled with increased computer capability and greater access to computers by the public, promises to give virtually everyone access to expertise. This will lead to profound changes in learning, in education and in our society as a whole.

REFERENCES

1. Gervarter, William; Expert Systems: Limited But Powerful, IEEE Spectrum, August 1983, pp. 39-45.

2. Kahn, Robert: A new Generation in Computing, IEEE Spectrum, November 1983, pp. 36-41

3. Martin James and Kleinlock, Leonard,: A Computer Industry Pearl Harbor, Technology Transfer News Vol. 1 No. 1. July 82.

4. Moto-oka, Tohru: The Fifth Generation, A Quamtum Jump in Friendliness, IEEE Spectrum, November 1983, pp. 46-47

5. Shane, Harold: The Silicon Age II: Living and Learning in an Information Epoch, Phi Delta Kappan October 1983, pp. 126-129

6. Torrero, Edward (Ed): Tomorrow's Computers, IEEE Spectrum, November 1983, pp. 34-35

TEACHERS: KEY TO COMPUTERS IN THE CURRICULUM

Vera T. Gershner, Ph.D.
Texas Woman's University, Denton

Johnny Lawton, Senior Evaluator
DISD, Dallas, Texas

Teachers are perceived as the most critical force in the classroom. Affective dimensions must be considered in order to facilitate the curriculum changes projected in Texas. Microcomputer acceptance and perceptions of micro use were assessed in research contrasting experienced teachers with a control group of teachers. Experienced teachers were in a CAI setting utilizing a mainframe computer. Significant differences between the groups suggest that teachers with little or no computer experience perceive a wide variety of uses and are accepting of new technology.

Issues of computer literacy bombard us...the need for it, definitions of it, and design of programs to build it. The mandated curriculum, the technology needs, the reexamination of schedules and space, and the impact of computers on learning and various subject matter fields all demand our attention. All of these are important, and now urgent. However, over the years, research has demonstrated regularly the overall importance of the teacher in the classroom.

Teacher attitudes, teacher knowledge, teacher behaviors, and teacher-child interaction have all been identified as significant in relationship to the development and learning of children. Today teachers are faced with a revolution in their professional world as well as their personal world. In selected states, of which Texas is one, the rate of change is intensified by state policies and mandates. Change is often painful, threatening, anxiety-producing, and stressful. So, as educational planners, administrators, or teachers, we must also focus on the human elements...the children and the teachers.

We have previously reviewed the literature in relationship to the attitudes of children, and other research is in progress directly studying children (Lawton & Gershner, 1982). The focus of this presentation is the classroom teacher, particularly those insights gained from research with groups of teachers. The experimental group includes teachers in a school which has utilized computer-assisted-instruction as an intrinsic part of the curriculum experiences for every child for more than seven years. The attitudes of these teachers are compared to an alternative group who also teach fourth, fifth, and sixth graders but who do not have comparable history of involvement with computers.

The Study

Research setting. This school of more than 600 pupils serves families which are predominantly black and frequently mobile. The student turnover rate is about 30% yearly. Therefore, a competency-based curriculum and technological assistance in managing instruction is particularly a propro.

Use of the computer. Each child at this "experimental" school has CAI as soon as he/she enrolls. Both the math and language curriculum are supported by computer programs.

The terminals are located in the library resource area and are supervised by a computer aide. The terminals are hard-wired to a mainframe which supplies the drill-and-practice programs and the management system for learning. Students are assisted by the computer aide who teaches the students to operate the terminals, maintains the schedule, and communicates results back to the classroom teacher.

Daily printouts reflecting the performance of each child are provided for the classroom teacher. She is encouraged to provide supportive experiences in certain areas where performance seems limited as it is reflected in the interaction with the computer program. When a child demonstrates 80% performance, he/she is automatically advanced by the program to another competency.

One of the strengths of the program appears to be the coordination of the selected CAI programs with the testing program of the District and the TABS test. This is particularly important in this magnet school.

Teachers in the experimental group. Sixteen teachers are assigned to this school. They are all experienced, black, and female. Although students move frequently, the teaching faculty is quite stable and appears to have a high esprit de corps. The principal is an active, dedicated, and enthusiastic instructional leader with intimate knowledge of the children, teachers, and staff. The teachers in this magnet school form the experimental group.

Teachers in the control group. Twenty teachers from diverse schools were utilized as a contrast group to the experimental group. They were all teachers of the fourth, fifth, or sixth grade. They were enrolled in coursework at Texas Woman's University, and they also completed the Teacher Attitude Survey (TAS).

Teacher Attitude Survey (TAS). The attitude instrument was adapted from Poirot"s attitude measure, and relates to perceived uses of computers and attitudes toward them. Among other things, the instrument was changed so that it specifically referred to microcomputers.

Background data on the teachers included grade level and computer experience and availability. The Likert-type scale asked for responses related to the degree of importance of various uses, and the degree of agreement for responses related to the attitudes. This instrument is self-administered, and was submitted anonymously by the subjects in the spring of 1983.

Data analysis. Utilizing the DEC20 at Texas Woman's University, the data were analysed utilizing ANOVA to explore differences in attitudes between the experienced and the limited-experience groups. An alpha level of .05 was established.

Findings

Perceived uses. The groups differed significantly on 11 of the 18 possible uses of the microcomputer. The contrast group attached significantly greater importance to: control of inventory, use with reluctant learners, management of schedules, budget management, assistance with basic skills, teaching of computer programming, use in wordprocessing, use in

grade reports, monitoring of competencies, attendance records, and development of computer literacy.

The groups did not differ significantly on the use of micros: with gifted/talented, with handicapped, with slow learners, for educational progress reports, for generation of instructional materials, for recording student files, and for problem solving.

Acceptance and valuing of micros. The groups differed significantly on 9 of the 17 factors relating to attitudes toward, and values of, microcomputers. The contrast group evidenced higher levels of agreement with statements related to: wanting to use micros more, wanting micro inservice, value of micros in assisting instructors, wanting micros for their work, belief in the micro as a motivating factor, wanting micros in the classroom, and the value of micros in managing the paperwork of the teacher. The experimental group placed higher value on buying micros ahead of other resources, but also indicated that micros were a luxury.

Areas in which there were no significant differences between the groups included: recognition that micros can improve education quality, the belief that teachers should know about micros, the sense that micros are not a threat to teachers, the belief that all students should be able to utilize micros, the belief that micros can be useful, ease of use, and discounting the idea that micros dehumanize or have adverse effects.

Teacher differences. Background factors revealed other differences between the two groups. The experimental group all experienced CAI, while only 50% of the contrast group indicated that computers were available in their districts. Only 6.3% of these experimental teachers owned a computer, and 25% indicated an intent to buy. The contrast group indicated that 40% of their students owned them, and that 50% of the teachers themselves intended to buy a computer soon. More teachers in the contrast group (35%) had attended an inservice than had teachers in the experimental group (19%). In both groups few teachers owned their own micros: 10% of the contrast teachers and 6.3% of the experimental teachers. A relatively low percentage of both groups had used computers in university studies; 20% of the contrast group and 12.5% of the experienced group.

Conclusions

Teacher attitude data relating to utilization of microcomputers have not yet been reported in our literature to the degree that one can contrast studies. Overall the first impressions from these data analyses appear to indicate that experienced teachers are more negative toward microcomputers than are the teachers in the contrast group. The contrast group attached higher levels of importance to many more of the items explored and agreed at higher levels with many of the attitude and value statements.

However, it is important to recognize that the long established CAI program which the experimental group experienced was a time-share type linked to a mainframe. Current trends in school utilization are directed to microcomputers and to a much wider variety of uses of the computer, i.e., classroom management, inventory control, student records, drill-and-practice, problem solving, simulation (Gleason, 1981). The literature now reflects provocative new uses of the microcomputer (Olds, 1983; Peelle, 1982).

A plethora of articles relate to staff development, computer literacy, and the planning of inservice and university training programs. Few articles deal with affective dimensions (Clement, 1981). The experimental group in this study to a large degree (81.5%) had not attended an inservice on computers. Also their organizational structure placed the terminals outside their classrooms and the management of that curriculum experience was assigned to an aide. It is not unreasonable to speculate that the identified differences are related to the organization of the experimental school.

As one type of check on this concept, the contrast group were compared to their cohorts who were teachers at other grade levels. This second contrast group included 58 teachers who were also enrolled in graduate studies. There were no significant differences between the two contrast groups.

Implications

The technological advances that brought computers to the school have now brought microcomputers to the classrooms. It is a new world, if one can only look and see. Without an informed, enthusiastic, and learning teacher, how can students grasp the dream and potential of the future? In some cases, and this study reflected one such case, an administrative decision brought the computers to the school. Other approaches include long term planning including the teachers (as recommended by our Educational Service Centers) or gradual exposure in diverse settings (Lee, 1983). Familiarity or propinquity does not necessarily build acceptance.

The term "computer literacy" may have outlived its usefulness had it not been built into our mandate; definitions and redefinitions abound (Inskeep, 1982; Luehrmann, 1981; O'Donnell, 1982; Levin, 1983). However, we know that it means far more than experiencing a drill-and-practice program (Anderson et al, 1981). We know that it means harnessing the power of the computer to work for us and to solve our problems. It provides a constant reminder of the need to dream, to plan, to experiment.

In Texas, we are charged with expediting the development of our children in a high tech world. In meeting that mandate, let us think carefully together about those most important people, the teachers.

References

Anderson, R., et al (1981). In defense of a comprehensive view of computer literacy. Mathematics Teacher, 74 (9), pp. 687-90.

Clement, F. J. (1981). Affective considerations in computer-based education. Educational Technology, 21 (4), pp 28-32.

Gleason, G. T. (1981). Microcomputers in education: The state of the art. Educational Technology, 21 (3), pp 7-18.

Inskeep, J. E. (1982). Computer literacy: What it is and why we need it. Curriculum Review, 21 (2), pp 138-41.

Lawton, J. & Gershner, V. T. (1982). A review of the literature on attitudes toward computers and computerized instruction. Journal of Research and Development in Education, 16 (1), pp 50-55.

Lee, H. (1983). How to lure teachers to the microcomputer. Principal, 62 (3), pp 26-27.

Levin, D. (1983). Everybody wants "Computer Literacy," so maybe we should know what it means. American School Board Journal, 170 (3), pp 25-29.

Luehrmann, A. (1981). Computer literacy...what should it be? Mathematics Teacher, 74 (9), pp 682-86.

O'Donnell, H. (1982). ERIC/RCS: Computer literacy, Part I: An overview. Reading Teacher, 35 (4), pp 490-494.

Olds, H. F., et al (1983). On understanding computers. Learning, 11 (8), pp 30, 34, 39, 42, 46, 50.

Peelle, H. A. (1982). Computer metaphors: Approaches to computer literacy. World Future Society Bulletin, 16 (6), pp 9-16.

COMPARING LOGOs

Dee Goldberg
Teacher, 5th Grade
Spring Branch ISD

Frank F. Matthews
Associate Prof.
Univ. of Houston / CL

This paper is a report on a pair of informal studies. The first was conducted with 5th graders and compared the implementation of the LOGO language on four different computer systems. The second was conducted with college students and compared three different implementations of LOGO on the Apple II computer system. The results of the first study showed a distinct advantage for two of the systems. The second study found distinct differences but no clear superiority.

Beginning in March of 1983, an informal study was conducted with a group of 55 fifth grade students at Terrace Elementary School in the Spring Branch Independent School District, Houston, Texas. The purpose of this study was to investigate which versions of LOGO, on which computer systems, were preferred by this group of students. The study was used as a part of the decision process for the implementation of LOGO instruction in the district. Since the class was broadly representative of a range of students including minimal English speakers and a range of socioeconomic classes the results should be of interest. In the Fall Semester of 1983, a class in LOGO was taught at UH/CL. This class was on the Apple IIe system and the students used the Apple, Krell, and Terrapin versions of LOGO for that machine. This paper is a combination of these two sets of comparisons.

The four sets of equipment and programs used in the elementary study were as follows:

Apple II+ (48K)
- Disk drive with controller card
- Language card (16K)
- Color monitor
- Apple LOGO on disk

Commodore 64 (64K)
- Disk drive
- Color monitor
- Commodore LOGO (Terrapin Corp.) disk Beta test version

TI 99/4A
- Peripheral expansion (32K)
- TI Color Monitor
- TI LOGO Cartridge

Radio Shack Color Computer (16K)
- Color Television
- COLOR LOGO Cartridge (Micropi Inc.)

Due to the time limits for this study only graphics aspects were implemented. Other aspects such as list processing, sound generation, and sprites were not considered. Apple, Commodore, and TI offer list processing. Commodore and TI offer sound generation, Radio Shack and TI offer sprites.

Color was an important aspect of LOGO to the students. Apple, Commodore, and TI offer a wide range of colors while the Radio Shack system uses only 4.

During this study, the students spent approximately 45 minutes a week working with the four versions of LOGO. Prior to the beginning of the study the students had learned LOGO on the Apple II+ system. At the beginning of the study, students were introduced to the other three dialects and were given access to the manuals and instructions for all four versions. LOGO lessons taught focused equally on each implementation of LOGO. Differences were discussed and student written procedures were modified to run on each of the four systems.

Students were encouraged to discuss the various implementations of LOGO and to share ideas. It was evident during the study that the students were evaluating the systems as well as the implementations of LOGO.

The students adapted very well to the different systems and rarely became frustrated with the differences that they found.

The students were eager to participate in the study and had a part in planning it. It was decided that an instrument was needed to evaluate the various implementations of LOGO, so a checklist was prepared. The students participated in the design of the checklist and many of the questions on the final version were initiated by them. The students were familiar with all of the questions used prior to completing the questionnarire at the end of the study.

After working for 10 weeks with the four different versions of LOGO, the students were asked to complete the 21 - item questionnaire for each systems implementation of LOGO. It was answered on a 5 - item Likert scale from strongly agree to strongly disagree. In each case the students were encouraged to add comments at the end.

The "way the turtle looks on the screen" was an important item for the students. The students liked to be able to see which way the turtle was headed and they liked the turtle to draw "clear, thin lines". The Apple and Commodore had the highest resolution and were preferred overall. The resolution on the TI was satisfactory while the Radio Shack robot was not a favorite. Many of the students commented negatively on the way that this turtle appeared on the screen. They also found the width of the lines drawn by it to be unsatisfactory. "The lines are too fat!" was a frequent comment about their turtle's track. On the other hand the students enjoyed the "Doodle" keys in this LOGO version. Doodle keys gave the students a simple single keystroke version which allows even nonreaders to draw with LOGO. An overlay is used to identify the number keys in the Doodle key program. One student participating in the study was just learning English. He enjoyed the Doodle key program and drew very effective designs and pictures with it.

A comparison of the mean responses for the LOGO instrument shows that the systems are divided into two distinct pairs. The Apple and Commodore LOGOs were much better received than the Radio Shack and TI. For 11 of the 21 questions the lower of the Apple or Commodore (A/C) mean response was higher than the higher of the Radio Shack or TI mean response (R/T). In all cases but Item 7 the lower (A/C) score was at least as high as the higher (R/T) score. The differences within the pairs were not as clear. The Apple was slightly prefered being higher

on 6 scores while the Commodore was higher on 2. The pair was, however, matched on 13 of the questions. The TI had a similar advantage over the Radio Shack with TI higher on 6 here while Radio Shack was higher on 3.

The differences within the pairs were not meaningful and may have resulted from small differences between the equipment as much as from inherent differences in the LOGO systems as presented. In particular the Commodore as stated above was a pre-release version and was not fully developed or documented. Also the TV on the Radio Shack system appeared to be very difficult to adjust.

The differences between the pairs are more difficult to explain and may be more inherent. Some ameliorating influences should be noted. Both of the lower rated systems were cartridge based and the students could not save procedures from session to session. Both of their display screens were more difficult to use with the TV color hard to distinguish and the TI monitor smaller than the others. The Radio Shack system was intended to be a smaller implementation on a much smaller machine. The TI suffered greatly from the restriction of the content to exclude sprites which could well have proved to be a major compensation.

Many students were proficient at writing procedures and drawing designs which took a considerable amount of space. All the implementations considered had a very restricted user workspace. Students generally did not find this to be a problem with the Apple and Commodore. The students were quite frustrated with the TI version of LOGO because the turtle often ran out of space before a procedure could be completed. The lack of usable space was the most frequent comment given about the TI LOGO. "I hate it when the turtle runs out of ink.", was a repeated comment.

Saving work was an important feature of LOGO to the students. The Apple and the Commodore both had disk drives and the students enjoyed saving and retrieving their work. The TI and the Radio Shack Color Computer used cartridge versions of LOGO and students were unable to save their work. The students were not pleased with that aspect.

The next work, comparing the LOGOs on the Apple considered three different implementations: Apple LOGO, Krell LOGO, and Terrapin LOGO. The course was conducted during the Fall Semester of 1983 and was intended to teach the LOGO programming language and to explore both its power and the effectiveness of the various implementations. For a variety of reasons, primarily associated with delivery and difficulties with making archival copies for protection, the three systems were not comparably utilized. In particular the Terrapin was used only minimally but there was sufficient experience to conclude that it was fully equivalent to the Krell implementations with noticeable differences only in their respective utility disk packages. Some of these could prove significant, particularly the machine language portions, but we lacked sufficient time to explore them properly. There were somewhat more differences between these two implementations and the Apple LOGO one. These differences proved to be surprising and in the reverse direction of what we had expected from reading the literature. The summary impression from the students was that the Krell/Terrapin LOGO was easier to use while the Apple LOGO was slightly more powerful. There were a variety of syntax differences between the two particularly in the procedures to set colors and read turtle conditions. The ease of use judgement

was primarily based on a somewhat easier edit process for the Krell/ Terrapin LOGOs. The slight power advantage to the Apple LOGO was for the Property List feature. The autoload feature of the Apple LOGO was also of some potential use in instructional settings but not a great deal.

The major difference between the implementations was the size of problem that they were capable of running. To make a fair comparison a procedure was created to define and run similar procedures until the system capacity was reached. The procedure was able to be defined using primitives common to all three implementations. In this test the Terrapin and Krell implementations again were completely equivalent stopping on the 48th iteration of the process. The Apple Corp. implementation, however, quit on the 24th iteration. Thus the Terrapin/Krell implementations would appear to have roughly twice the capacity of the Apple implementation.

The differences between the systems were far overshadowed by their joint limitations and advantages. The major limitation experienced by the students was the inability to define and use disk data files. This was one of two obstacles to the solution of any programming problem in LOGO. The other is the somewhat restricted size of available workspace which limited the size of program. Both of these restrictions should be ameanable to solution on systems of 128K as opposed to the 64K for current ones. Beyond this, all three implementations tried were found to be fully recursive procedural languages each capable of utilizing all of the usual features of structured programming. In addition the ability of the language to make a program self-modifying while preventing the development of a full compiler provides a significant increase in range.

QUESTIONNAIRE

1. I think that learning LOGO has helped me learn to use the computer.

2. I think the books that come with this LOGO are easy to read.

3. I think the editor in this LOGO is easy to use.

4. I like the way the turtle can draw lines of many lengths, in this LOGO.

5. I like the way the turtle draws quickly.

6. I think circles are easy to draw with this LOGO.

7. I like the way this LOGO tells you if you are out of space.

8. I like the way the turtle draws diagonal lines, in this LOGO.

9. I think the colors in this LOGO are clear and bright.

10. I like the way the screen looks with this LOGO.

11. I like the background colors in this LOGO.

12. I like the pen colors in this LOGO.

13. I like the way the screen looks with this LOGO.

14. I like the way the turtle shows up brightly in this LOGO.

15. I like the size of the LOGO graphics screen.

16. I like the way the turtle responds in this LOGO.

17. I like the way the Control keys are used in this LOGO.

18. I like the way the computer keeps the commands in order on the screen.

19 I like the way I can save my work with this LOGO.

20. I like the way I load this LOGO.

21. I like using this LOGO.

MEAN RESULTS BY SYSTEM

	App	Com	TI	RS
1 —	1	1	1	1
2 —	1	2	2	2
3 —	2	1	2	2
4 —	1	1	2	2
5 —	1	1.5	2	2
6 —	2	3	4	3
7 —	2	3	3	2.5
8 —	2	1	2	2
9 —	1	1	2	5
10 —	1	1	2	2.5
11 —	1	1	2	4
12 —	1	1	3	4
13 —	1	1	2	3
14 —	1	2	2	3
15 —	2	2	2.5	2
15 —	1	1	2	2
17 —	2	2	2	2
18 —	1	1	2.5	2
19 —	2	2	3	3
20 —	1.5	2	2	2
21 —	1	1	2	2

NASSI - SHNEIDERMAN DIAGRAMS - AN ALTERNATIVE TO CLASSICAL FLOWCHARTING

Ranette H. Halverson
Instructor of Computer Science
Midwestern State University, Wichita Falls, Texas 76308

This paper presents an intorduction to the use of Nassi-Shneiderman diagrams, an alternative to basic flowcharting for program design. The emphasis is on their use by beginning students. The use with BASIC language is considered.

As an instructor of beginning computer science students, I have found that teaching a student a programming language is futile without adequate instruction in program design. The typical attempt has been through classical flowcharting. One problem is that it easily lends itself to the production of "spaghetti code."

A number of techniques have been proposed to produce "better"programs: pseudocode and HIPO charting to name two. However, the technique to be discussed here is that of Nassi - Shneiderman diagrams. These have also been called block diagrams due to their appearance. A N-S diagram is a large block made up of smaller blocks, each representing one or more instructions.

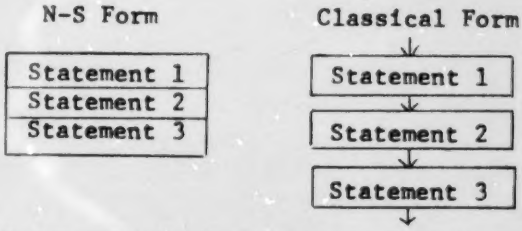
The intent of this paper is to present an introduction to N-S diagrams with beginning programmers in mind, and the problems they encounter.

The steps recommended to the students are as follows:

1: Begin with a basic program module of the correct form.
2: Clearly distinguish each individual task to be performed. Imbed these tasks within the program module in the appropriate order. Continue adding detail until it is in codeable form.
3: Code and debug each module until satisfactory results are obtained.
4: (For more advanced students this step is added.) Compact the program to a more efficient form.

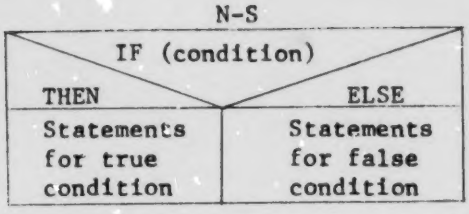
While these steps may be implemented with any design technique, we shall use the basic N-S structures. Let us first look at the basic structures and compare them to the corresponding classical flowcharting forms. The basic programming structures, often considered the minimal set, include SEQUENCE, IF-THEN-ELSE, and DO-WHILE. Others which may be added as the language allows are CASE and REPEAT-UNTIL.

(1) SEQUENCE BLOCK: The sequence block contains a single statement, such as I/O, arithmetic, assignments, and subroutine calls.

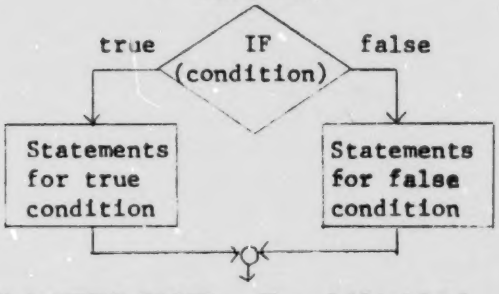


(2) IF-THEN-ELSE BLOCK: The if-then-else block contains one or more conditions and two alternatives, true or false. Each alternative may contain zero or more statements.

N-S

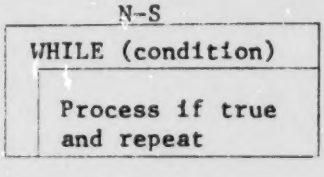


Classical

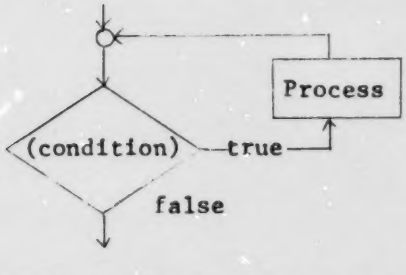


(3a) WHILE BLOCK: The while block indicates one or more statements to be repeated as long as the stated condition remains true.

N-S

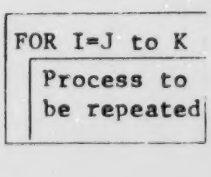


Classical

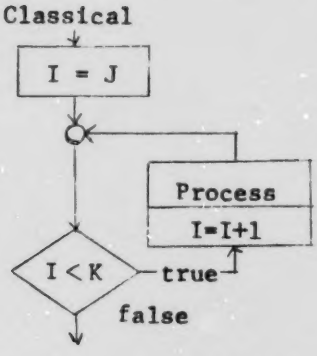


(3b) ITERATION BLOCK: Since most languages have an iteration, the FOR in PASCAL or FOR/NEXT in BASIC, etc., the while block may also be used for these.

N-S



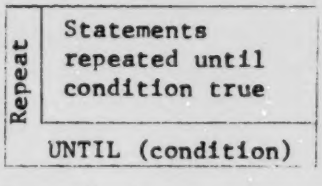
Classical



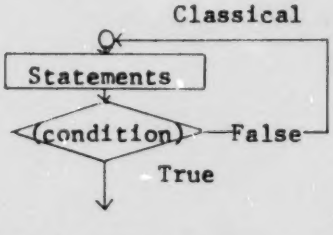
The above 4 structures will allow us to solve any problem, particularly those encountered in beginning programming courses. There are, however, two other structures which are convenient when the language being used has the corresponding instructions.

(5)REPEAT-UNTIL BLOCK: The repeat-until block indicates one or more statements to be repeated while the condition is false. That is, continue until the condition is true then stop. This is the opposite of the While-Do which continues while the condition is true.

N-S



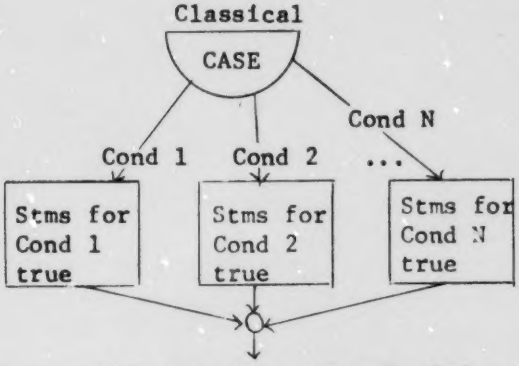
Classical



(6) CASE-BLOCK: The case block indicates a choice among several alternatives. It is not recommended for less than 4 choices. (Note that the implementation of the case statement is very language dependent.)

N-S

CASE			
Cond. 1	Cond. 2	...	Cond. N
Stms. for Cond 1 true	Stms for Cond 2 true	...	Stms. for Cond N true



To exemplify the steps recommended to students, let us proceed through a typical programming problem using the Nassi-Shneiderman forms.

As mentioned above, step 1 is to choose a basic program module. A program module is a very general N-S diagram of the problem which indicates the relationship of input, output, and processing of data. That is, it describes the nature of the program. For example, do we intend to input all data before beginning processing; do we need to input and process one data item at a time; when should the output take place; etc. There are many forms, but some of the most common are as follows:

(a)

Input All Data
Process All Data
Output All Data

(b)

While more data
input 1 item
process 1 item
output 1 item

(c)

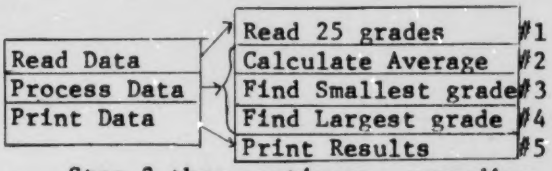
While more data
input 1 item
process 1 item
Output all data

The student may select one of these, or the more advanced students may develope their own. However, the intent is to be very general in nature.

Problem Description: Given a class of 25 students, each having 1 grade, we would like to calculate the class average, class range, and print these results.

Step 1: For this problem we shall use module (a). Shown above.

Step 2: We then distinguish the basic tasks to be performed and imbed them into the module. We determine that there are 5 of these tasks.



Step 2 then continues, expanding upon each task until the coding may easily be done. Two more detailed versions are shown.

#1	Read 25 grades into an Array
#2	Sum the 25 grades
	Divide Sum by 25
#3	Search array for smallest
#4	Search array for largest
#5	Print Average, Smallest, Largest

#1	FOR I=1, 25
	Read Array (I)
#2	Sum = 0
	FOR I = 1, 25
	Sum = Sum + Array (I)
	Avg = Sum/25
#3	Small = Array (1)
	FOR I = 2, 25
	If Small > Array (I)
	then / else
	Small = Array (I)
#4	Large = Array (1)
	FOR I = 2, 25
	If Large < Array (I)
	then / else
	Large = Array (I)
#5	Print Average, Small, Large

This is an easily codeable form of the N-S diagram. I do not wish to indicate that only the above 3 versions should be used. The programmer should use as many versions as is necessary.

Step 3: At this point we shall forego the actual coding since the diagram virtually contains code but will discuss how we could easily debug this and other problems. The first step would be to run the program and remove the syntax errors. Of course, we should check the results by hand to insure they are correct. For some problems we may even want to use a smaller set of test data to simplify the debugging.

It is advisable to debug each block, one at a time, by adding one or more intermediate print statements, typically to the end of the block. For this example, we could add to the end of each task:

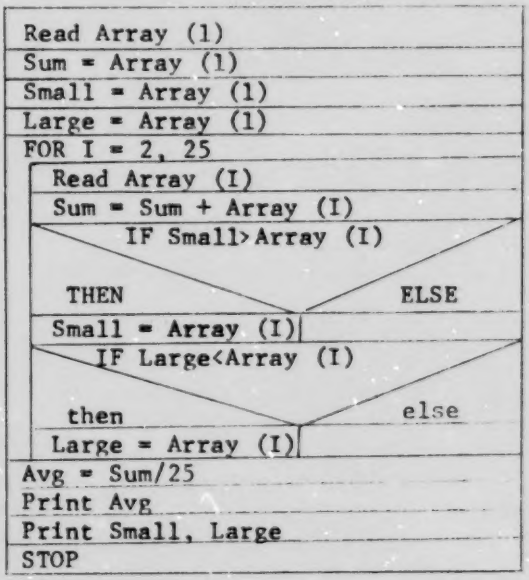
Task #1 - Print the array
#2 - Print the Sum and Average
#3 - Print Small
#4 - Print Large

These steps allow an orderly procedure through the program, getting closer and closer to the solution as each block is debugged. It is particularly advantageous to the student whose instructor is likely to give partial credit for demonstrating that some tasks do work properly. While the above intermediate print statements may seem trivial, I have found it is not uncommon for students to "destroy" previously correct variables before reaching the output stage.

As problems become more complex more intermediate write statements must be used. Then as tasks are debugged these print statements may be removed.

Step 4: It is of course obvious that this is not the most efficient form of this program. However, for beginning students a working program of any form is an important milestone. For those more advanced, or even some beginners we shall now compact the program into a more efficient form. Notice, that we do this only after we have a working program.

There are many possibilities for compacting the above program. One of them might be to combine tasks #3 and #4 in order to search through the array once. We may compact further and combine all 3 processing tasks into one. We could even combine all 4 interations into one, which is shown below. Keep in mind, however, that compacting the program tends to hide the logic and the basic tasks. In many cases this also confuses the student.



Let us now address the issue of coding. In a structured language such as PL1, PASCAL, or FORTRAN 77, the coding is almost trivial, since the basic N-S structures are a part of the language. However, in an unstructured language such as BASIC which is prevalent among the micros, the coding is not trivial, at least not to a beginner. Thus, I have attempted to construct some pseudostructures or general forms which BASIC students may use. The justification for constructing these forms for the student is to continue to avoid spaghetti code and minimize the use of GOTO statements.

Using the following forms the student simply fills in the line numbers and at each "bracket" fills in the applicable statements. If indention is allowed, then it is recommen-

ded for the "bracket" statements.

CREATING AN IF - THEN - ELSE IN BASIC

```
   IF NOT (CONDITION) THEN GOTO XX
   REM "THEN" CONDITION
           {
           {
           {GOTO YY
XX     REM "ELSE" CONDITION
           {
           {
YY     REM "END IF"
```

IF - THEN - ELSE WITH NULL ELSE

```
      IF NOT (CONDITION) THEN GOTO YY
      REM "THEN" CONDITION
          {
          {
YY    REM "END IF"
```

EXAMPLE:

IF X < Y	
THEN	ELSE
PRINT X	PRINT Y
Y=X+Y	X=X+Y
READ X	READ Y
STOP	

IN BASIC

```
 10     IF NOT (X < Y) GOTO 90
 20     REM "THEN" CONDITION
 30        PRINT X
 40        Y=X+Y
 50        READ X
 60        GOTO 130
 90     REM "ELSE" CONDITION
100        PRINT Y
110        X=X+Y
120        READ Y
130     REM "END"
200        STOP
```

CREATING WHILE BLOCK IN BASIC

Since there is no while statement in BASIC we shall have to create it using the if - then statement.

```
      REM "WHILE" STRUCTURE
XX    IF NOT (CONDITION) GOTO YY
          {
          {
          {(STATEMENTS TO BE REPEATED)
          {GO TO XX
YY    REM "END WHILE"
```

EXAMPLE:

WHILE X > 0
READ Y
PRINT Y
X = X *Y
PRINT X

```
100   REM "WHILE" STRUCTURE
110   IF NOT (X> 0) GO TO 160
120      READ Y
130      PRINT Y
140      X=X * Y
150      GOTO 110
160   REM "END" WHILE
200   PRINT X
```

This paper is a similar, but condensed version of the presentation I have made to my students. I have used N-S diagrams with absolute beginning programmers up through junior and senior computer science majors and have found it to be very successful in all instances.

References:

(1) FORTRAN 77 Language and Style, Michael J. Merchant, Wadworth Publishing Co.

(2) Encyclopedia of Computer Science and Engineering, Second Edition Anthony Ralston, editor.

(3) Nassi-Shneiderman Charts - An Alternative to Flowcharts for Design, Computer Programming Management, 1979. Auerbach Publishers

RANDOM SELECTION ALGORITHMS FOR COMPUTER-BASED INSTRUCTION

William J. Harvey, Ph.D.
Associate Professor of Computer Science
Director, Instructional Systems Laboratory
East Texas State University
Commerce, TX 75428

ABSTRACT: Methods of obtaining random integers in BASIC, MUMPS, PILOT, Pascal, and APL are described. Programming methods for using random integers obtained to select data for CAI tasks are presented. Algorithms are described which deliver random selection of tasks in conjunction with such features as inhibiting of repeats, feedback on items missed, repeating of missed items, and efficiency in obtaining random order.

The random selection of data is a vital component of the design of much computer-based instruction. If a large group of students are to use a given lesson, or if the same student benefits from reuse of a single module, lessons incorporating random selection features meet instructional needs better than lessons employing a single linear sequence.

Let us first examine the practical question of how random integers can be obtained in programming. Examples and discussion will deal with programming environments used for CAI: BASIC, MUMPS, PILOT, Pascal, and APL. Then we can explore design considerations which affect implementation options.

Random selection builds on a pseudo-random number generation feature normally available in the programming language. In BASIC, for example, you can invoke pseudo-random number generation by using a function RND, usually with a single argument: RND(0). Consult the particular manual regarding argument options. If you are using Level II BASIC or Radio Shack MicroPILOT for the TRS-80, you have two options:

```
1)BASIC> RND(0) returns a real
  number between 0 and 1
2)BASIC> RND(<integer>) returns
  an integer between 1 and the
  integer given as an argument
```

To exploit this feature in your program, assign the result to a variable:

```
3)BASIC> X = RND(0)
  or  X = RND(4)
4)PILOT> C:X=RND(0)
  or  C:X=RND(4)
5)MUMPS> S X=$R(4)
6)APL>   X←?4
```

Some environments will provide you with only an integer or only a real number. Notice that standard MUMPS offers only the integer result. The symbol "?" in APL is called "roll." Notice that the TRS-80 features given as examples (from Level II BASIC and MicroPILOT) return an integer between 1 and the integer (inclusive), while the $RANDOM function in MUMPS returns an integer between 0 and the integer minus 1. Random number generation is normally based on sequences resulting from repetition of a certain calculation, where a seed value is passed on from one calculation to the next. In Applesoft, a negative argument will "reseed" the process, beginning a new sequence. On Commodore 8032 and 64 computers, RND(0) causes generation of a value based on three clocks, rather than on a sequence of calculations. If you are using TI Extended BASIC on the TI 99/4A, the RANDOMIZE command can be included in your program to assure that a new sequence of calculations is

initiated each time your program is run. On any system where there is no prepared feature to "reseed" the sequence of random number calculations, you can experiment with obtaining a single clock value and utilizing a digit from it to provide a new seed value. If you are using Pascal, you will have to design your own function or procedure for random number generation, as standard Pascal does not include such a feature. Random number generation functions are presented by Grogono [4] and Miller [7].

For random selection, you normally need an integer value result. In order to convert an real number between 0 and 1 into a usable integer in the range you desire, multiply the real number by the number of integers in the range desired, truncate the product, and add the intended lower bound of the proposed range. For example:

TO OBTAIN A NUMBER					
BETWEEN	4	22	-2	6	1
AND (inclusive)	20	122	+2	60	100
MULTIPLY BY	17	101	5	55	100
AND ADD	4	22	-2	6	1

```
7)BASIC>  X =INT(RND(0)*17)+4
8)Pascal> X := TRUNC(RANDOM(SEED)*17)+4; {assuming user's function}
```

To exploit the series of random integers obtained by methods such as those described above, a generalized frame can be designed which needs question data (cues) and response evaluation data (correct answers or matches) according to a standard format. If we choose a simple example, imagine a CAI module which quizzes users on the capitals of states of the United States:

```
QUESTION:  What is the capital
  of [........]?
```

For operation of such a module it would be necessary to store the name of a state and the corresponding name of the capital city for each state to be included. The table of state names and city names could be stored in DATA statements in BASIC (as an internal sequential file), in a sequential file on disk or tape, in a random access or relative file on disk, or in a string array (after having been read into the array from DATA statements or a file).

As teachers who produce their own software often find DATA statements convenient, let us examine the method for random selection for a set of DATA statements as shown in code segment (9).

We could call this the "musical chairs" algorithm; when the loop reaches the value stored in N, random state and city names will remain in the question (cue) and answer (match) variables. Where random access file records are being used, it is merely necessary to use the random integer obtained to identify a record.

Random integers obtained by the methods shown above may also be used to select labels in BASIC, MUMPS, and PILOT. In BASIC, the random integer can be assigned to a variable which is then evaluated by an ON...GOSUB statement as a multiway branch to subroutines. Indirection and postconditionals in MUMPS offer great flexibility in automating multiway branching and modification of the flow of control. Some versions of PILOT permit evaluation of string expressions to determine the label for a J (jump) command. Integers can be converted to strings and, if necessary, concatenated with other strings, to form acceptable labels. In other versions of PILOT the option of using conditional expressions

```
9)BASIC>  Assume data is stored as follows...
          10000 DATA STATENAME1,CITYNAME1,STATENAME2,CITYNAME2...

          480 REM SELECT NUMBER FROM 1 THRU 15 INCLUSIVE
          490 N = INT(RND(0)*15)+1
          500 REM MAKE SURE DATA POINTER IS PROPERLY (RE)SET
          510 RESTORE
          520 REM LOOP N TIMES (ACCORDING TO INTEGER OBTAINED)
          530 FOR K = 1 TO N
          540 REM READ ANOTHER PAIR (STATE AND CITY)
          550 READ Q$,A$
          560 NEXT K
          570 REM USE Q$ TO ASK QUESTION; A$ FOR MATCH AGAINST INPUT
```

(instead of just the Y and N conditions) makes it feasible to exploit random integers in J and U (use subroutine) command lines.

Now that we have examined how to obtain and exploit random integers, we can turn our attention to random selection algorithm alternatives in CAI. The design of computer-based instruction utilizing a random selection feature requires attention to the following questions:

1) May the tasks be repeated during the same session?

2) When a student responds unsuccessfully to a task, should that task be repeated during the same session?

3) Are all of the tasks of the same level, or does the lesson include tasks of various difficulties?

4) Does the learner receive a report on particular items missed on conclusion of the lesson or subsequently?

The algorithms presented below are intended to provide the tools for control of random selection in accordance with the objectives and data sets of a lesson design.

Inhibiting repeats of data in a single CAI session was originally approached by using an algorithm in which flags were set and checked. A random integer was obtained by methods such as those described above, and if the flag corresponding to the integer chosen had been set, then another random integer was obtained and the checking process repeated. This type of algorithm was recommended for Coursewriter by Cody and Milstein [2] and more recently for BASIC by Hennefeld [6]. The disadvantage of such an algorithm is that normally only a fraction of the random integers generated are actually usable. Serious delays can occur as repeated attempts are made to locate an unflagged integer, especially after most of the allowable integers have already been selected.

Inefficiency in random integer selection can be avoided by the following algorithms:

Bennett [1] proposes an in-place exchange: an array of the size of the number of integers needed is initialized with the respective integers. A loop is established to operate for the number of iterations as the number of elements in the array. In each iteration, the integers stored in the elements identified by the current loop counter value and the random integer obtained are exchanged (code segment (10)).

```
10)BASIC>
    100 REM INITIALIZE ARRAY
    110 DIM A(15)
    120 FOR K=1 TO 15: A(K)=K: NEXT K
```

```
480 REM LOOP TO EXCHANGE VALUES
490 FOR K=1 TO 15
500 N = INT(RND(0)*15)+1
510 T=A(N): A(N)=A(K): A(K)=T
520 NEXT K
```

The result will be an array containing indices in random order. Only one array is required, which can be an advantage if a large amount of data is required. As the instructional session proceeds, indices stored in the array are used without being altered. Therefore, if a flag array is added in which flags are set for missed questions, it is possible to obtain a list of missed questions for learner feedback or teacher report when the session is concluded.

Heilborn and Talbott [3] and Orwig [8] have proposed variations of a "shrinking list" algorithm. Heilborn and Talbott's scheme calls for initializing an array with integers as described above for Bennett's algorithm. The multiplier for random integer selection is initialized. The contents of the array element identified by the current random integer are displayed or used. Then from the position used to the one preceding the final position in the array, each element is assigned the value of the succeeding element in order. Then the multiplier value used in obtaining random integers is decremented by 1. If the multiplier value still contains a value greater than 0, then the process is repeated. Heilborn and Talbott call this a "shuffle." Orwig provides an example of a CAI module which similarly eliminates array elements from consideration as the process of random selection proceeds, but does so by exchanging the selected and lowest elements in the array in combination with decrementing the multiplier. In Orwig's module, the data to be used in instruction are stored in a string array. In a companion dimension, flags are stored for use in marking items missed and to be repeated. If the stored answer and received answer match (correct input), the multiplier is not decremented and the appropriate repeat flag is set. The question thus remains in the pool of those which can be selected, but is marked so that the learner can be informed that the item is a repeat when it comes up again. If correct input is received, the data stored in the lowest element of the array still within the scope of the random integer selection process is transferred to the position of the data just utilized. The multiplier is then decremented, and if the multiplier is greater than 0, another question is presented. Of course, this algorithm is destructive to the contents of the array and preserves, in the form given, no record of which items are missed. However such a feature could be easily added.

Turner [9] includes an interesting algorithm in a proposed program "Create a Quiz" for the TI 99/4A. Davis' book on programs for the TI 99/4A [3] makes extensive use of randomization for lessons and games.

A random selection algorithm used at East Texas State University, locally called the "unsort" algorithm, resembles algorithms described above. An array and multiplier are initialized as for Bennett's and Heilborn and Talbott's algorithms. Similarly as in Orwig's algorithm, the scope of the array is reduced by transferring values not yet selected into a smaller range of contiguous positions. If it is not necessary to use the array contents more than once in the same program without reinitializing it, or it is not necessary to refer to it for feedback or records, the contents of the current bottom element can simply be transferred to the most recently used position whenever the most recent random integer is not equal to the multiplier value. Notice that in contrast to the method used by Orwig, here the value selected is an index to data located elsewhere. A second array would be required to preserve the integer sequence obtained by this method.

```
11)BASIC> 100 REM ARRAY P IS INITIALIZED WITH INDICES IN ORDER
          105 REM DATA STATEMENTS ARE NOT SHOWN HERE
          110 REM ARRAYS Q$ AND M$ WILL CONTAIN CUE AND MATCH
          120 Q = 15
          130 DIM P(Q),Q$(Q),M$(Q)
          140 FOR N=1 TO Q: READ Q$(N),M$(N): P(N)=N: NEXT N
          150 REM COUNTER TO BE DECREMENTED IN Q2 - Q IS PRESERVED
          160 Q2 = Q

          500 FOR N = 1 TO Q
          510 Y = INT(RND(0)*Q2)+1
          520 X = P(Y)
          530 P(Y)=P(Q2)
          540 REM 'X' USED AS INDEX TO ELEMENT IN ARRAYS Q$ AND M$
          550 Q2 = Q2 - 1
          ......{presentation, match, feedback}.....
         1000 NEXT N

12)MUMPS>
  UNSORT  ;AMT IS NUMBER OF QUESTIONS - B IS ARRAY TO BE UNSORTED
          S AMT=15
          F L=1:1:AMT S B(L)=L
          S A=AMT
          F L=1:1:AMT S R=$R(A)+1,T=B(A),B(A)=B(R),B(R)=T,A=A-1
          Q
```

Code segments (11) show use of the "unsort" algorithm in its simple form. Code segment (12) exhibits a variation on this algorithm in the form of a MUMPS module or routine which exchanges the values so that a complete unsorted array results, stored in the original array. The multiplier value being successively decremented effectively partitions the array into "used" and "unused" portions in case of interruption.

For those who have access to APL for developing of CAI, the mixed primitive function "deal" can be used to obtain an "unsorted" array:

```
13)APL> A←15?15
```

The first argument determines the number of elements in the array and the second, the range of random integers to be selected.

Where tasks of different levels of difficulty are being presented in the instruction, it is necessary to store level codes in association with the data items (cue and match) for each item. If these are grouped, then the random integer selection can be adjusted according to provide adaptation to the appropriate level of difficulty required for the learner.

REFERENCES

[1] Earl Bennett, "Computers in Education: Randomizing Lessons," an unpublished paper (Greenville: E-Systems, 1982).

[2] Ronald P. Cody and Barney M. Milstein, Instructional Strategies Utilizing a Computer Based Instructional Language (White Plains, NY: IBM, 1976), pp. 42-43.

[3] Steve Davis, Programs for the TI Home Computer (Dallas: Davis, 1983), pp. 65-72.

[4] Peter Grogono, Programming in Pascal, Revised Edition (Reading, MA: Addison-Wesley, 1980), pp. 117-118.

[5] John Heilborn and Ran Talbott, Your Commodore 64: A Guide to the Commodore 64 Computer (Berkeley, CA: Osborne/McGraw Hill, 1983), pp. 150-154.

[6] Julien Hennefeld, Using BASIC: An Introduction to Computer Programming (Boston: Prindle, Weber & Schmidt), pp. 127-128.

[7] Alan R. Miller, Pascal Programs for Scientists and Engineers (Berkeley, CA: Sybex, 1981), pp. 29-35.

[8] Gary W. Orwig, Creating Computer Programs for Learning: A Guide for Trainers, Parents and Teachers (Reston, VA: Reston, 1983), pp. 83-88.

[9] Len Turner, 101 Programming Tips & Tricks for the TI 99/4A Computer (Woodsboro, MD: ARCsoft, 1983), pp. 38-39, 55-56.

The English Learning Center System

M. Hawkins, M. Nigliazzo, V. Stone, W. Demaree, C. Sphar, H. Tomlinson
Del Mar College
Corpus Christi, TX 78404

The English Learning Center System is a comprehensive study course in the basic skills of English grammar, punctuation, and usage implemented using microcomputers. The System has evolved over five years into a body of printed courseware in continuous use at the English Learning Center at Del Mar College. Now the System is being computerized. It is also available in text form as The English Learning Center (Kendall/Hunt, 1983).

History: The English Learning Center System began some five years ago as a body of printed courseware. During that period, it has been refined through continuous use by many hundreds of community college students at all levels of skill. The System was developed as a self-paced course in which students could focus their work on their own specific areas of need and take their own paths through the body of materials. Although this system was originally prepared in paper form without computers in mind, it is not simply "programmed instruction." The computer just makes a great deal of complex branching and decision-making transparent to the student and allows his teacher to monitor his progress more accurately, as well as giving the student instant feedback. The process of computerizing the System began in 1982. The System has also been published in book form by Kendall/Hunt as The English Learning Center, 1983.

Content: The System consists of nineteen topic units covering all areas from basic sentence structure and grammar to subject-verb agreement, punctuation, and diction. (See the list below.) Each topic unit has three levels of difficulty, from basic to challenging, making it suitable for students in the upper elementary grades through early college.

Topic Units:

1 - Parts of Speech
2 - Verbs
3 - Sentence Parts
4 - Adjectives & Adverbs
5 - Phrases & Clauses
6 - Sentence Sense
7 - Commas
8 - Commas & Semicolons
9 - Semicolons, Colons, and Dashes
10 - Subject-Verb Agreement
11 - Pronoun-Antecedent Agreement
12 - Reference of Pronouns
13 - Case of Pronouns
14 - Misplaced Modifiers
15 - Dangling Modifiers
16 - Apostrophes
17 - Capitalization
18 - Quotation Marks and Underlining
19 - Diction

How a Student Works Through the System: On beginning a unit, the student first takes a Diagnostic Test at the level he (or his teacher) has chosen. Depending on his results, he may be moved to another level (lower or higher), where he again takes a Diagnostic Test.

If he stays at the same level, he moves toward mastery of the material by studying a set of Review Notes and then

taking a sequence of Trial Tests (for which he is not graded). The real teaching takes place in the Trial Tests, where the student gets brief trial hints (those keyed to the number of tries he makes) and longer triggered hints (a particular expectable--but erroneous--answer by the student triggers a preset hint on the screen). These longer hints display selected portions of the Review Notes pertaining to specific errors the student makes, allowing for on-the-spot teaching of specific subject matter keyed to the particular student's demonstrated needs. On most units, he even has a "help" function available to him as he ponders his answer to a question. In Subject-Verb Agreement, for example, by typing "R" (for Review) instead of the question's answer, he gets a brief bit of review material from the Review Notes.

His performance on the Trial Tests may take him back to the Review Notes for further study, or he may proceed to the Proficiency Test sequence. There are two Trial Tests and two Proficiency Tests at each level. The student's results on the two Proficiency Tests may be used to determine his completion of the unit or his need for further study.

On the final Proficiency Test, the student may return to the menu for another topic unit, move to a higher level of his current topic unit, or return to the Review Notes. At this point, it is up to the teacher monitoring his progress to decide whether his Proficiency Test scores constitute passing the unit or not. (The teacher may set the order in which the student works through any or all of the nineteen units. Most students work straight through in sequence.)

The materials in the tests and in the Review Notes become succeedingly more comprehensive and challenging as the student moves up from level I toward level III. New material is introduced in the Review Notes of higher levels and covered in the testing there. Level I material is suitable for upper elementary and junior high school students and remedial secondary and college students.

Free-Standing Test Modules. In addition to the nineteen subject-area units, the System includes three test modules, one for each of the three levels of difficulty. Each test module consists of the Diagnostic Tests for all nineteen units in the level, arranged in six focal groups. For example, one group contains tests for commas; commas and semicolons; and semicolons, colons, and dashes. A student choosing this group would get a score for each subgroup and an average for the group. These free-standing test modules allow teachers to administer a range of Diagnostic Tests quickly to individual students or entire classes. They can also be used as final mastery tests, taken after the student completes the Proficiency Tests of several topic units. In this case they allow teachers to test in a more comprehensive way the work that the student does in a series of individual lesson units. With the testing done separately, the student may feel less inhibited working on the individual lesson units. And, without the optional apparatus for collecting student scores, the test modules become free-standing practice tests for the entire range of lesson units at each of the three levels.

Preparation and Use of the System. These units were prepared for the computer with Radio Shack's AUTHOR I Lesson Development Program. Each unit consists of a number of "pages" (screens) of instructional material, questions, and branching options to move the student to other parts of the unit or the system as a whole, depending on his test responses (his test results are used by the computer to select a pathway to the student's next work). The student takes the course, executing his chosen unit, using AUTHOR I's TEACH program. In all cases, the student is informed of his score on a test and of the next action to be taken in his particular

sequence. He also receives brief, clear instructions about his options in paging through the text and selecting answers to questions. The System's instructional clarity is remarkably good.

The nineteen units in the English Learning Center System may be used one at a time on a single computer or on a Radio Shack computer network with a hard disk.

THE THINKING OF FIVE-YEAR-OLD LOGO PROGRAMMERS

Sandra N. Hines
2614 Robinwood
Denton, TX 76201

Five-year-old children are capable of programming in LOGO. They define graphics problems by drawing pictures on paper. Then they attempt to give the Turtle commands which will make the same designs on the monitor. If the commands they give do not produce the desired result, kindergartners may display behaviors which are typical of cognitive development for their age group. Individual problem-solving strategies and programming styles of young children are discussed.

Procedures of the Study

This paper reports the programming behavior of five kindergarten children who participated in a research study during the fall semester of 1982. The purpose of the study was to ascertain whether five-year-old children could perform simple programming in the LOGO language. Programming was defined as the ability to give a series of commands in response to a problem, name them, and recall the program by name at a later time.

The three girls and two boys who participated in the study were randomly selected from a pool of ten kindergartners who passed a pretest of numeral and capital letter identification. The researcher worked individually with each child every day for about thirty minutes on an Apple II Plus microcomputer for ten weeks. Each session was tape recorded and transcribed. The data was later sorted into categories of behavior.

At the conclusion of the study, two tests of programming competence were administered. The first test consisted of a series of four dot tasks of increasing complexity. Participants were asked to connect the dots on the monitor by giving the correct commands. The second test required students to write or dictate procedures which would make the LOGO Turtle draw three shapes: a square, a rectangle, and a triangle. These procedures were recorded on paper first and then tested on the computer. Subsequently revisions of their written programs were made by the children if needed. Both tests of programming allowed students to demonstrate their knowledge of number quantity, in distance and angle estimates. The second test was designed to show whether the participants understood the structure and sequential order of a procedure. Children were asked to locate the errors or "bugs" in their test procedures independently.

Results of the Programming Tests

The programming tests revealed the problem-solving stategies used by the children as well as their competence in LOGO programming.

Dot tasks. Three students completed the four dot tasks successfully. One student completed three of the tasks, and one succeeded only on two tasks.

Four students used a cautious strategy that moved the Turtle in small increments until it reached the dot. These students all exhibited preferences for particular numbers in choosing the inputs. Four of the students used a PEN ERASE, BACK strategy to restore the Turtle to its former position after errors were made.

One student displayed a good understanding of number quantity. Her number inputs were accurate estimates. She was able to add numbers and used this as a problem-solving strategy. She would give smaller numbers as inputs, add them, and use their sum as an input. Her other logical strategies included giving a large estimate and inching the Turtle to the dot with smaller numbers and the reverse: giving a large estimate and reducing the inputs by small amounts until the Turtle reached the dot. For instance, she tried the following sequence of FD inputs: 70, 65, 64, 63, 62, 61.

The last dot task required students to turn the Turtle at a 45^{0} angle. Students who were most successful in this task tried repeatedly, up to five times, with different inputs to get it turned correctly.

Shape programs. Students were asked to write procedures for a square, a rectangle, and a triangle. Two students wrote workable procedures for the three shapes. Two students wrote successful procedures for two shapes, and one student wrote only one correct shape procedure.

All students experienced difficulty with the triangle problem. Three students tried to make a triangle with an angle of RT 90. They were accustomed to using this to make turns in their other programs and did not understand why it did not work in a triangle. They did not make any attempts to change the procedure. The two students who made triangles recognized that RT 90 would not work. One of them stated, "Now we have to get a bigger number (than 90)." The other tried RT 89 and kept adding RT inputs until it turned enough for a triangle. (RT 120 is the angle needed for an equilateral triangle.)

Programming Styles

In the daily programming sessions, the kindergartners defined their programming tasks by drawing pictures on paper of the design they wanted the Turtle to draw on the monitor. They decided which part of the picture the Turtle should draw first, second, third, and so forth. Some students chose to write separate procedures for each part of the picture. These were eventually combined (in the order in which they were written) into one program which drew the picture on the monitor. Other students preferred to write one continuous procedure.

Responses to Incongruity

There were some instances in which the children were not successful in directing the Turtle to draw what they had planned on paper. When they were faced with such a situation, students exhibited one of the following responses.

1. They did nothing. They did not show any outward signs of recognizing that the design on the monitor differed from their picture. This response could be attributed to fatigue or to not realizing that the two were different. In his causality experiments, Piaget observed that young children had difficulty noticing contradictions. They tended to see the aspects that were the same as those they had planned and ignored any discrepant elements. Many of the paricipants displayed this reaction during the daily programming and during the programming posttests.

2. They denied that the result was their fault. Sometimes students refused to accept responsibility for the graphics on the monitor. They blamed the Turtle when the result was not what they had planned. One girl scolded the Turtle: "Now you Turtle! Whey don't you straighten up?"

3. They liked the unexpected result. They modified their original plan to incorporate the novel elements.

4. They disliked the unexpected result. Sometimes students rejected the appearance of unwanted elements in the picture on the monitor and typed in the reverse to eliminate them. Then they would try again to make the computer graphics conform to their plan on paper.

Summary

Five-year-old children were able to program in LOGO. They displayed individual differences in their programming styles and in their problem-solving strategies. All of the participants were able to locate and fix some errors in their posttest procedures. However, at times the children showed no response whatever when there was a definite difference between the desired goal and the result on the computer monitor. This puzzling response may be due to the developmental level of the participants in the study. The response of transferring blame for the undesired result is also typical of the thinking of young children.

Kindergarten children are capable of a surprising degree of computer programming competence. However, adults must realize that young children may not perceive or respond to some problems in a "logical" manner because of their developmental immaturity.

IMPROVING WRITING IN THE PRIMARY GRADES THROUGH COMPUTER TECHNOLOGY

James L. Hoot, Ph.D.
College of Education
North Texas State University
Denton, Texas 76203

This paper explores current research which suggests word processors may have a great deal of potential for improving writing as a communication process in the early elementary grades. The author concludes with work still needing to be done to make computers more effective accouterments of primary grade writing programs.

Over the past decade there has been a great deal of interest in improving writing skills among students at all levels. Much of this concern has resulted from reports such as the National Assessment of Educational Progress which strongly indicted American schools for not producing students with better writing skills. "American schools," according to the report, "have been successful at teaching students to formulate quick and short interpretations, but have not yet developed in students the skills they need to explain and defend the judgments they make." The study also found that students performed well on multiple choice items, but were very poor at expressing themselves in essays (NAEP, 1981).

There are many possible reasons for students' apparent lack of writing ability. Perhaps most noteworthy, according to a recent investigation, is the way writing is being taught in the schools (Applebee, 1981). In an extensive study conducted at the secondary level the researcher found subjects spent 44% of their time engaged in processes relating to writing. Fifty percent of that time, however, was composed of menial tasks such as filling in blanks or writing simple response answers. Moreover, he found that only 3% of the subjects produced one or more paragraphs during class time and only 3% of homework time required students to write even one paragraph. Nearly a third of the teachers participating in the study indicated they NEVER required their students to write even a simple paragraph.

At the primary level writing as a communication skill is given even less attention. Although instructional time allocated to writing varies from school-to-school, one need only observe a few "writing" sessions or speak to a few students or teachers to realize that writing is most often operationally defined as "handwriting" or technical skill development rather than as a process whereby students express their important ideas through composition.

We live in an age where improved communication is essential for our survival. Yet, unless our educational system begins to look upon writing as a means of expressing MEANING the writer wishes to communicate rather than a method of improving technical aspects of print, improvement appears unlikely. The paper which follows will focus upon recent research into

the origins of writing which suggests word processor computer technology may have a great deal of potential for improving writing abilities of very young children.

Writing and the Young Child

The back to basics movement has resulted in two major shifts in early childhood writing research. First, there has been a shift from an emphasis upon the written product (e.g. handwriting skills) to a focus upon the process used by children to become proficient writers. Secondly, interest in the origins of writing has shifted attention to how younger children acquire the skills of writing (Harste, Burke, and Woodward, 1981, Hiebert, 1978, King, 1981, Deford, 1980, Vukelich and Golden, 1984).

Perhaps one of the most extensive reviews of research on children's writing to date (Vukelich, 1981) concluded that effective beginning writers engage in three major processes: (1) prewriting (preparation for writing and includes thinking through the task at hand) (2) composing (the actual writing as well as making decisions about what goes where, stopping and starting, and rereading) and (3) rewriting (altering, clarifying, and refining ideas). From a research perspective, those who teach writing by stressing only the mechanics of writing rather than structure an environment conducive to the three major processes of writing, do a disservice to their students.

Word processors, specifically designed to minimize writing mechanics, are beset with problems for the very young children (e.g. too complex for beginning writers, limited typing abilities of primary children, etc.). Nevertheless, the technology offers tremendous potential for facilitating a more conducive writing environment which allows instruction to focus more upon preparation, composing, and especially rewriting.

Word Processors and the Improvement of Writing

Future prospects for improving writing instructions are poor unless more effective approaches are developed from the earliest grades onward. One such approach made possible by the increasing availability of microcomputers in primary classrooms is the use of word processors. The capability of rapidly manipulating lines of text through insertion, deletion, and movement of passages within a composition, of being able to easily correct errors, of being able to store text for later revision, and of being able to rapidly print out polished copy suggests that word processing has great potential for improving written communication by alleviating the technical drudgery of writing.

Research concerning word processors and primary grade children is nearly nonexistant. Yet a few studies conducted with middle and upper elementary students are now available. A recent report entitled, "Hugh Potential Seen in Largely Undeveloped Field: Sophisticated Microcomputers Used to Teach Students to Write" (Toch, 1982), indicated that: "Microcomputers when programmed as word processors--which require typing skills of a student rather than the simple ability to respond with a yes or no answer to questions on the computer screen--are a potential boon to students' writing and thinking skills." Similarly, evaluative data reported by DiGammario (1981) indicates that positive benefits result from use of a word processing program at the elementary school level. Findings suggest that students using the word processor revise their work more, have less fear of having to produce new copy, write longer stories, follow directions,and pay attention to detail.

Conclusion

Research reported herein suggest that word processors may have a great deal of potential for improving writing instruction in even the earliest of grades. Yet, much work remains in order to maximize this potential in actual classrooms. First of all consumers will have to demand more simplified word processing programs from the software industry. Even the simplest of current programs are much too difficult for first grade students. Secondly, teachers will be forced to rethink the relative value of formal handwriting instruction. It may be more educationally appropriate, for example, to devote a portion of time allocated to handwriting to TYPING instruction. Finally, if teachers and children are to profit from the computer's ability to improve writing instruction, guidance must come from research with primary grade children.

References

Applebee, Arthur N. WRITING IN THE SECONDARY SCHOOLS, Urbana, IL: National Council of Teachers of English, 1981.

Deford, D. "Young Children and Their Writing." THEORY INTO PRACTICE 19 (Summer, 1980), p. 157-162.

DiGiammarino, F. "Text Editing." CLASSROOM COMPUTER NEWS, (November/December) 1981, p. 32.

Harste, J., Burke, C., and Woodward, V. CHILDREN, THEIR LANGUAGE AND WORLD: INITIAL ENCOUNTERS WITH PRINT. Final Report. Washington, DC: National Institute of Education, 1981.

Hiebert, E.H. "Preschool Children's Understanding of Written Language." CHILD DEVELOPMENT 49 (December 1978), p. 1231-1234.

King, M., and Rentel, V. HOW CHILDREN LEARN TO WRITE: A LONGITUDINAL STUDY. Final Report. Washington, DC: National Institute of Education, 1981.

National Assessment of Educational Progress, "Reading, Thinking, and Writing: Results from the 1979-80 National Assessment of Reading and Literature," National Assessment Distribution Center, Suite 700, 1860 Lincoln St., Denver, CO 80295.

Toch, T. "Hugh Potential Seen in Largely Undeveloped Field: Sophisticated Microcomputers Used to Teach Students to Write. EDUCATION WEEK 2 (February 2, 1982), p. 13.

Vukelich, C. "The Development of Writing in Young Children: Review of the Literature." CHILDHOOD EDUCATION 48 (January/February, 1981) pp. 167-170.

Vukelich, C. and Golden, J. "Early Writing: Development and Teaching Strategies." YOUNG CHILDREN 39 (January, 1984), pp. 3-8.

COMPUTER MATHEMATICS

John Huber
Mathematics Department
Pan American University

ABSTRACT

Computer mathematics is not computer programming. Several outlines of topics appropriate for a computer mathematics class will be examined based on various mathematics prerequisites for the computer mathematics course.

Computer mathematics as used in this paper is a mathematics course in which the computer is used to develop, enrich or reinforce mathematics concepts and principles. It is not a programming course in which the examples and problems are chosen from mathematics. A course in which computer programming is the major objective has a place in the high school curriculum but it should not be used as a substitute for a mathematics course. This paper will examine various mathematics topics in which the computer can serve as a vehicle for their development.

With only one year of algebra as a prerequisite, number theory provides a wealth of mathematics topics for exploration. One can begin with such topics as: divisors of a number; sum of divisors; abundant, deficient and perfect numbers; prime numbers; units and composite numbers; prime factorization; and generating prime numbers. Each of these topics has various levels of mathematical sophistication which can be developed. In addition, such topics as greatest common divisor, least common multiple, the Euclidean algorithm, Pythagorean triples, modular arithmetic and the use of modular arithmetic in scheduling can be studied.

With geometry as a prerequisite such topics as calculating pi using Archimedes' inscribed polygons; calculating pi using harmonic-geometric means; classifying a triangle as scalene, isosceles or equilateral; classifying a triangle as acute, right or obtuse; calculating areas of triangles and other polygons using determinants; determining collinearity of points using determinants; and determining if a point is on, in the interior of, or in the exterior of a triangle. Again the development of the mathematics of each of these topics is the primary objective.

With second-year as the prerequisite the following topics are appropriate: evaluating polynomials using a nested algorithm; synthetic division; calculating roots using the se-

cant method, the bisection method and the fixed point method; roots of a cubic equation, Gauss elimination method; Gauss-Jordan algorithm; calculating permutations and combinations; binomial expansion; and Pascal's triangle. In addition, the following topics in probability and statistics are appropriate calculating the mean and variance of a set of data using the two-pass algorithm and the update algorithm; simulations of rolling a die, binomial experiments, and geometric distribution experiments; Monte Carlo methods for pi; and Buffon's Needle Experiment for pi. The mathematical level of each of the above topics is at a level accessable to students with two years of high school algebra.

The above list of topics is merely suggestive and not exhaustive. In addition, not all of the above topics can be covered in a one-year course in computer mathematics if the appropriate mathematics is developed. It is also essential that testing in such a course be on the mathematics and not on the programming.

The computer can be an exciting teaching aid for the learning of mathematics. However, in the mathematics classroom it should be just that, a teaching aid. The computer mathematics course offers us an opportunity to explore mathematics using the computer but we must remember we are teaching mathematics and <u>not</u> computer programming.

INTERFACING AN OPTICAL READER WITH A MICROCOMPUTER

John Huber
Mathematics Department
Pan American University

ABSTRACT

This presentation will demonstrate and describe how a Scantron Optical Mark Reader can be interfaced with an Apple microcomputer. A discussion of the development of a program to analyze the Pre-Assessment of Basic Skills test using the Scantron will be included including a discussion of the algorithms for decoding the information from the Scantron.

An Optical Mark Reader (OMR) is one method of enter-information into a computer. The purpose of this paper is to examine how the Scantron can be interfaced with an Apple microcomputer to provide student, teacher and district results on the Pre-Assessment of Basic Skills test package.

The Pre-Assessment of Basic Skills test package is a series of criterion-referenced test in mathematics and reading for grades one through nine. Each test containg four items per objective. For example, the grade nine mathematics test containg fourty-four items on eleven objectives.

The first step in the program is to enter descriptors for each of the objectives. The descriptor of Objective 1 is "ADDITION AND SUBTRACTION WITH REGROUPING." This information is entered through the keyboard and stored in a file.

The econd step is to enter which items measure each of the objectives. Objective 1 on the grade nine mathematics test is measured by items 6, 16, 27 and 30. This information is also entered through the keyboard and the program then creates a file associating each item number with its objective.

Next the answer key is entered through the Scantron for each of the test and stored in a file. The descriptors, item-objective information and answer keys are entered only once for each test.

Once the above information is entered, each class is ready to be graded. The program begins by asking for the teacher's name, the grade level, class period and subject. Each student's answer sheet is entered through the Scantron and the information is stored in a file in alphanumeric form.

The scoring program then decodes the student's name, scores each question and determines if three or more items per objective are correct. The student's name and results for each objective are stored in a file. In addition, the relusts are updated to a district file for the specific grade level and

subject area.

A Diagnostic Profile is printed for each student including the student's name and a descriptor of each objective on which the student missed two or more items. A Class Report is printed for each class including the subarea, the teacher's name, the class period and a table of student name by objective with an "X" indicating objectives missed by each of the students. When all the tests have been graded, a District Report is generated including the percentages of mastery of each of the objectives by grade level.

BIOGRAPHY OF AN ELEMENTARY LIBRARY LEARNING CENTER - COMPUTER LAB

by Barbara Husting and Sandra Webb
Fort Sam Houston Elementary School
3370 Nursery Road
San Antonio, Texas 78218

---the history, development, present usage and future plans for the newly enlarged Fort Sam Houston Elementary Library Learning Center and Computer Lab.

In San Antonio, Texas, in September of 1952, Fort Sam Houston Elementary School opened its doors to 385 students in eighteen classrooms. Designed to meet the educational needs of the elementary age children of military families living on the post of Fort Sam Houston, home of the Fifth Army, Brooke Army Medical Center and the Academy of Health Sciences, the school has increased in enrollment to over 900 students in early childhood through sixth grade. Its physical plant has expanded to included 52 classrooms, a kindergarten section, and an all purpose building for physical education classes.

A diverse, highly mobile population from a broad spectrum of socio-economic, ethnic and cultural backgrounds is housed on the post and served by the district, which in 1961 opened the Robert G. Cole junior-senior high school to provide on-post educational facilities for the 7th through 12th grade pupils who previously had been bussed to junior-senior high schools in other districts. The present total enrollment of the district is now about 1,500 students (although it fluctuates daily with the general nature of military assignments and the particular nature of the short term courses offered in the medical training facilities on post).

The thrust of the educational program has broadened and been enlarged as the district enrollment and physical facilities have grown. This is in keeping with the philosophy of the Board of Trustees from which we quote in part "We acknowledge the fact that each student has needs and purposes related to personal living experiences and this contemporary relationship can be demonstrated, explained, and strengthened through the District's help," and "We believe that equal educational opportunity is the right of all students without regard to race, creed, color, national origin, ethnic background, sex, or handicapping

condition...." This philosophy led to the development of a strong special education program and one of the first gifted-talented programs in the State reaching students in all grades at the elementary level.

From the gifted-talented program came Fort Sam Elementary's introduction to computers in a school setting. With one TRS-80, a few cassette tape programs, the assistance of high school mentors in the gifted program, and the guidance of an ever-questing teacher, the gifted students experienced their first in-school use of computers in 1981.

In an effort to prepare teachers for using computers in the school, various staff members were sent by the district to conferences and workshops sponsored by Trinity University, South West Texas State University, University of Texas, Region 20 Education Service Center, the Texas Library Association and TCEA. The thrust of these conferences varied greatly, but our cumulative awareness of the history, philosophy of use, care and maintenance, and management of computers was broadened. The opportunity afforded by these conferences to preview software, make contact with sales representatives, and exchange ideas with others was valuable.

In 1981-82 using the knowledge gained from these staff experiences, two Apple 11+'s with disk drives, using RCA large screen monitors, were purchased by the school to be used in the gifted progam and for staff development. A committee of volunteers from each grade level and special areas began meeting with Keith Imon from Region 20 Education Service Center for inservice on computer literacy, history of computers, and educational uses of computers.

In 1982-83, five more Apple 11+'s were purchased along with seven Panasonic 12inch color monitors and two printers. Grades one through six and special education had computers for their classroom use. In assessing inservice needs at the beginning of the school year, the faculty and staff of the elementary school indicated interest in participating in inservice sessions about the microcomputers. Each faculty or staff member participated in eight one-hour sessions with Keith Imon using the seven Apple 11+'s.

The volunteer committee did not repeat instruction but was led by the librarian/media specialist in a search for appropriate software. An evaluation tool was selected, catalogs perused, reviews accumulated from all sources possible, sales representatives contacted, and software brought into the school for previewing purposes.

As we explored Region 20's materials, they requested some of our staff members to evaluate commercial software programs for service center purchase. This allowed some of our staff to preview larger curricular programs which in turn increased our awareness of available items and the need to refine our objectives in selection of programs to meet our needs.

Our initial software consisted of MECC programs offered through special

contract with Region 20. These programs were subject area based. From this beginning, we began to look for more software to balance grade level and subject area needs.

The instructional coordinator, who had been the gifted-talented teacher, was visiting other schools where gifted students were using computers. These visits led to other contacts to visit schools where computers were available for the general population use. Ideas were gained from these visits, but our plans were more extensive in program development.

As our knowledge and experience increased, the awareness that all students did not have access to microcomputers became apparent. With administrative leadership and Board commitment, plans were initiated to enlarge the existing library and construct a computer lab for all elementary students.

An architectural firm with a great deal of experience in school and library construction was selected. The librarian and administrators conferred with the architect as he drew plans to be presented to the Board for approval. A valuable tool used in making these plans had been prepared by the Learning Media Services Committee for the Southern Association Evaluation in 1980. At that time the committee had prepared a comparison study of the existing library floor space and the ALA recommendations for learning centers.

Although compromises had to be reached to stay within the budget approved by the Board, a design was selected which met most of the spatial goals for a learning center and a computer lab. Shelving footage and locations were worked out, special cabinets were designed to hold different types of non-print media, color schemes were determined, new furniture and equipment were ordered, and old furniture was re-arranged on paper.

As we moved from the planning stage to the construction stage, the library staff's real physical exertions began. Over 700 hundred boxes holding some 14,000 books and items of non-print media were inventoried, boxed, labeled. and moved out of the library to storage.

Although a number of students were involved in helping with the packing, we wanted the entire student body to be aware of why the library had to be closed and of the progress we were making in packing and moving. A contest was held in which the students guessed the total number of boxes needed to pack the library collection.

After packing was completed, the librarian and her aide found temporary work space on the stage where they processed new materials and worked with small groups of students. Life under the pipes and behind the curtains lasted for four months. When the interior of the center was close enough to completion, work began unpacking those 700 boxes. Volunteer assistance from parents and staff made the reshelving move more easily and swiftly.

When students entered the remodeled center for the first time, they discovered a spacious facility with ample shelving, a story-telling area, a reference section, an AV production room, a library office, an AV equipment storage area, and a conference room. The most fascinating aspect of all was the computer lab with its 28 microcomputers located at one end of the center.

We received TEA's House Bill 246 with the proposed Computer Literacy Framework and studied the essential elements, objectives and grade-level flowchart. In keeping with the framework, our Board of Education's educational philosophy, and our knowledge of students' interests and conceptual development, a computer literacy philosophy was developed which would allow our students to maximize their experiences.

The piecing together of our scope and sequence was undertaken in September of 1983. We again incorporated the essential elements of the TEA proposal and included a section on moral and ethic issues and consumer education skills. Because of the number of microcomputers which would be available with the purchase of new equipment and the regularly scheduled instructional time students would have on the computer, we placed some age-appropriate skills at a lower grade level than the TEA guidelines suggested.

The new equipment included purchase of 36 Apple IIe's. Twenty green screen monitors were purchased to allow for 80 character display and word processing. Sixteen color monitors would capitalize on effective presentation of commercial programs and allow students to explore graphics. An RF modulator was purchased to adapt a black and white television set so that teachers or students could demonstrate with a large screen for large or small group presentations.

Three of the new Apple IIe's were equipped with green screen monitors, dual disk drives and printers to be used in the library office, school office and computer lab. With those three and the 27 units in the lab, the remaining new six systems were placed on carts for classroom use.

Eighty-five percent of the student population would be coming to the computer lab once a week for a half-hour instructional experience. The students' interest and self-motivation with computers became apparent immediately. Initial paper and pencil activities with the keyboard arrangement provided an acquainting experience preceding actual use. Through the exploration and use of the LOGO language, students are able to master many of the objectives listed in our scope and sequence. The enlarged keyboard, constructed by sixth graders, has become has an effective teaching tool with all grade levels. News articles, advertisements, and cartoons have encouraged discussion and consumer awareness as well as reinforcing computer terminology.

While classes come and go to the computer lab, classes in library skills are in process. Through the cooperation of faculty and students, there is little disturbance of these different library activities.

The administrative and office staff are beginning to explore programs that would meet their needs. One of the counselors is using this computer to compile her monthly statistical report. Other faculty members are discovering programs that prepare classroom materials.

One intermediate teacher has been piloting the use of a grading program. After the beginning frustrating experiences and the extra time involved in the duplicate recording of grades on paper and on disk, he has been able to provide each of the 125 sixth graders with a three weeks' progress report or a six weeks' history of assignments and grades. The sixth grade students have been impressed with this fast grade-averaging system and it has also facilitated a student's understanding of the impact of a zero grade in grade averaging. Other classroom experiences include: using drill and practice exercises, being rewarded with use of simulation programs, or curricular related tutorial experiences.

As some of the faculty were new in 1983-84, it was decided to survey the entire staff's attitudes and self-assessed computer skill level in preparation for an inservice day devoted to computers. Workshops for groups no larger than 15 were planned to meet the needs as identified in the survey. The schedule was developed to include assigned morning workshops determined by need and elective workshops in the afternoon for the 80 staff and faculty members.

Workshops ranged from sessions for first time microcomputer experiences to guest sessions for LOGO, Apple Writer, Bank Street Writer, and reviewing the previous year's experiences with BASIC. One main session was held to present and discuss the computer literacy philosophy statement and scope and sequence. All rooms, outlets, and equipment were utilized to allow for these sessions. A very successful seesion provided faculty time to preview our own software collection and perusal of other new materials. Time for relaxed discussion of software, curriculum needs, student usage and orientation to the library/learning center was appreciated by all.

Teachers were made aware that all software was cataloged like other non-print media, shelf list maintained by the librarian, and other entries added to our card catalog for the school population to use. Shelving of the software is also in the main library. Catalogs and computer magazines are housed in the computer lab next door as we continue to search for appropriate software materials to meet our needs.

We have been trying to include the community in our development. The Newsletter with grade level and special areas messages goes home once a month. The primary and intermediate open houses provided hands-on experience for many parents and allowed students to demonstrate newly acquired skills.

We have had guests from the news media and Army leadership. Our Congressional representative visited our new facilities. We have shared program development and the lab with another elementary school for their inservice session. Our new instructional

coordinator brought her UTSA class as they explored technology and education of the future.

In our immediate future at the elementary school, the computer lab will move into a new room next to the library learning center. The librarian plans to use the vacated space for listening and viewing centers. The new room will serve as the permanent home for the lab. Adjacent to it will be the Basic Skills Lab.

We plan to continue our search for quality software to be used within our curriculum. We are exploring CMI programs that provide student testing, record keeping for faculty, for the office and library.

We would like to explore more varied use by students of the computer lab at other than instructional time. We will begin piloting small groups of kindergartners in the lab during this next month. We have had requests from parents for beginning experiences with computers. We hope to meet these requests in the near future as well as encourage other adult usage.

We have discovered that educators involved with computers are most willing to share successes as well as problems, and even admit lack of knowledge. We wish to expand our networking with other resources - both human and material.

A DISTRICT PLAN FOR COMPUTER LITERACY
Design and Implementation

E. Gillette Irby
Corpus Christi Independent School District
P.O. Box 110
Corpus Christi, TX 78403

A district plan should be designed to insure the success of an instructional program for computer literacy. Goals, objectives, and plans for operation and instruction must be developed. Three models for instruction may be considered. Instruction may be provided by infusing appropriate objectives into the existing curriculum, by units of study in a "pull-out" program, and/or by special courses. Staff development must support the needs of staff members in fulfilling their expected roles in the program.

There is a rapidly increasing need for students and educators to become computer literate. Curriculum planners are baffled by the need to implement instruction to meet the great demand. They are asking, "What is Computer Literacy?" "How does one become computer literate?" "What is the nature of an acceptable program?" This paper suggests some considerations and possible options.

It is necessary to have a district plan for computer literacy in order to provide guidelines for central staff administrators, campus administrators and teachers. The plan should include a working definition of computer literacy which provides for the focus of a program with activities to meet the needs of students and teachers. A general definition might read as follows:

> Computer Literacy is whatever a person needs to know and do with computers in order to function in his/her role in an information based society.

In this case, a program could be designed to provide for an individual to elect those objectives which would provide for his/her needs.

The plan should include a goal which states a time that a specified population will have acquired specific computer-related skills as listed in the plan.

There should be a description of the operation of the plan. Will there be a K-12 program? If so, will it be coordinated at each of the following levels: elementary, junior high school and/or senior high school? Will there be a phase-in of the plan? For what length of time will plans be made--short range and long range? Short range plans might be for one to two years, while long range plans might be for three to five years.

The instructional objectives for computer literacy must be determined and a model designed to accommodate them. These objectives should include the essential elements for computer literacy proposed by Texas Education Agency (TEA) and the local district's scope and sequence. At least three different models may be considered for implementing the objectives. They are the infused model, the unit model, and the course model.

The objectives might be achieved through infusing them into the content of the

regular curriculum. This requires that a decision be made as to which course content can most effectively absorb the computer literacy objectives. A careful study of the proposed TEA essential elements for computer literacy shows that they could appropriately be placed at various grade levels in language arts, social studies, mathematics, and science. The objectives could be infused, that is embedded, into the regular content. Thus, the objectives would be achieved without supplanting any course content.

The infused objectives by subjects suggested could be as follows:

Language Arts
. computer related vocabulary
. word processing
. topics for composition

Social Studies
. history of computing
. social impact
. legal issues
. problem-solving with simulations
. data base management

Science
. the physical design for the computer
. precise language
. data organization
. hypothesis testing

Mathematics
. logic
. program design
. mathematical tasks like computations, iterations, and graphs

A unit model could provide for objectives to be organized into specific units to appropriate instruction to be implemented in a "pull-out" program or as special instruction in an existing course. These units of any desired length might be developed on selected topics such as

. History of Computers
. Advantages of a Computer
. Programming Languages
. Introduction to Programming.

Special courses might be designed for summer school and/or the regular school term. In these courses instruction could provide for the achievement of computer literacy objectives. Examples of such courses are

. Computer Awareness
. Introduction to Programming
. Computer Literacy (Grade 8)
. Computer Literacy (Grades 10-12)
. Word Processing
. PASCAL
. Data Bases

Employee development is an essential component to the successful implementation of a computer literacy program. Administrators and their instructional staffs must have the computer-related competencies necessary to the responsibilities of their respective jobs. A plan should provide for a phase-in and ongoing program. It should have a two-fold purpose:

. To teach appropriate levels of computer literacy competencies
. To provide instructional techniques and strategies for teaching computer literacy in the classroom.

The employee development program should be based on

. The Texas Education Agency's Essential Competencies for Educators
. The district's goals for the computer literacy instructional program
. An employee needs assessment.

Training seminars, primarily planned for regular inservice days during a phase-in period, should be for administrators and instructional personnel who

. Have had little or no training in computer literacy

. Need specific assistance in the utilization of the computer in the designated priority subject areas
. Need assistance in computer programming.

The program should be designed with the recognition that there are differences in the levels of competencies needed. The principal should assess the employee development needs of his/her faculty to ensure that each faculty member teaching computer literacy objectives has appropriate training for the teaching assignment. This is necessary for an equitable implementation of the computer literacy program for students throughout the school.

A special four-week minicourse for educators may be desirable. A center with a qualified instructor and appropriate equipment could be established. Teachers and administrators desiring to extend their skills on computers could enroll for three 3-hour sessions each week for four weeks. Such a course can likely qualify for Adult Vocational Training for the purpose of extending required job related skills.

College and university courses, conferences, and computer-corporation-sponsored workshops could provide additional opportunities for training instructional personnel.

Other Considerations

Other considerations in developing a plan for computer literacy include the following:

. Software and hardware acquisition
. Maintenance
. Security
. Budget.

In summary, a model for computer literacy instruction must include:

. a definition of computer literacy
. instructional objectives for computer literacy
. a goal for achievement
. an operational plan
. a plan for employee development
. a plan for software and hardware acquisition
. a maintenance plan
. a plan for security of equipment and software
. a plan for the evaluation of process and product
. a budget.

The implementation of a model for the successful achievement of computer literacy in a school requires the commitment of the community, the Board of Education, administrators and the instructional staff.

BIBLIOGRAPHY

Anderson, Ronald, et al. "In Defense of a Comprehensive View of Computer Literacy-A Reply to Luehrmann," Mathematics Teacher, Vol. 74, No. 9, December, 1981, 687-690.

Bergin, Victoria. "Computer Literacy: Who? What? Why? When? Where?" TASB Journal, September, 1982, 33-34.

Bitter, Gary G., "The Road to Computer Literacy: A Scope and Sequence Model," Electronic Learning, September, 1982.

_____. "The Road to Computer Literacy, Part II: Objectives and Activities for Grades K-3." Electronic Learning, October, 1982.

_____. "The Road to Computer Literacy, Part III: Objectives and Activities for Grades 4-6," Electronic Learning, November/December, 1982.

_____. "The Road to Computer Literacy, Part IV: Objectives and Activities for Grades 7-9,"

Electronic Learning, January, 1983.

_____. "The Road to Computer Literacy, Part V: Objectives and Activities for Grades 10-12," Electronic Learning, February, 1983.

Cupertino Union School District, "K-8 Computer Literacy Curriculum," The Computing Teacher, March, 1983.

Levin, Dan. "Everybody Wants 'Computer Literacy,' So Maybe We Should Know What It Means," American School Board Journal, Vol. 170, No. 3, March, 1983, 25-28.

Luehrmann, Arthur. "Computer Literacy: What Is It; Why Is It Important?" Electronic Primer, May, June, 1982, 20-22.

_____. "Computer Literacy-What Should It Be?" Mathematics Teacher, Vol. 74, No. 9, December, 1981, 682-686.

Rogers, Jean B. An Introduction to Computing: Content For A High School Course. International Council for Computers in Education, University of Oregon, Eugene, Oregon, 1981.

Texas Computer Education Association and Texas Association For Educational Data Systems. Position Paper on Computer Literacy, 1982.

Texas Education Agency. Guide For Selecting a Computer-Based Instructional System, TEA, Austin, TX, 1982.

The Texas Education Agency, Essential Competencies for Educators, TEA Austin, TX, 1982.

MAKE IT MOVE !

BY ELLEN JOSLIN

MCULLOCH MIDDLE SCHOOL
HIGHLAND PARK I.S.D.
DALLAS , TEXAS

Many programming skills can be most easily taught through the use of graphics. The desire to animate computer images results in a "need to know" which promotes learning. The "print-erase-move-print" method for animating text, low resolution graphics and high resolution graphics pictures is discussed.

Pac Man moves. Students come to computer class knowing computer pictures can be animated. They want to know how to "make it move". Too often teachers need to teach things (necessary things) that boys and girls are less than enthusiastic about learning. It is easier and pleasanter to teach a child something a child very much wants to learn. Boys and girls willingly spend hours putting pictures on a computer screen. Making these pictures move provides a satisfaction not often found in the classroom. The child creates something tangible and uniquely his/her own. Creating and building tangible things occurs all to seldom in most classes and for many students is a very gratifying experience.

It is also a valuable learning experience. Some might argue that students working with graphics get little experience with "real programming" and will lack comprehension of important programming techniques such as decisions , loops, and structured programming. I have found that a solid foundation of BASIC programming concepts can be taught by giving boys and girls experience with graphics , and that there is a carry over ; students are able to apply those skills to more traditional programming problems. The children want to learn what the teacher needs to teach. There is a "need to know" which is often lacking in the traditional curriculum. Properly channeled, this motivation promotes learning.

Pictures in motion can be achieved without intoducing the Apple Computer's graphic mode. The text mode program below will simulate a rocket take-off. It will also introduce the child to loops. I call the computer TOM , and tell the children that TOM stands for Totally Obedient Moron. The computer is very strong and very stupid. It will do exactly what you tell it to do,and it will do it over and over again. There is always some child who will change the program below to make the computer print a hundred rocket ships. Of course , they fly by too fast to count. That is the ideal

time for someone to suggest that we let the computer count the rocket ships. It is easy to slip in a line 95 which tell the computer to PRINT N.

```
5  REM ** This program
illustrates motion  achieved in
text mode
10 HOME
20 FOR N = 1 TO 10
30 PRINT "    /\    "
40 PRINT " /     \  "
50 PRINT " ¦     ¦  "
60 PRINT " ¦     ¦  "
70 PRINT " ¦     ¦  "
80 PRINT " ¦     ¦  "
90 PRINT " -^ ^-   "
100 NEXT N
110 END
```

Allow the boys and girls time to experiment with different pictures and with different size loops. Let then discover that the motion achieved with this method is always vertical and never horizontal. Kids will eventally be unhappy with this limitation.

You might then wish to introduce the "print - erase - move - print" method of animation. Horizontal motion is achieved by the program below.

```
5 REM **This programm
illustrates the "print - erase
- move - print"  method of
animation
10 HOME
15 C = 1
20 FOR N = 1 to 39
30 HTAB C + N
40 SPEED = 50
50 PRINT "->";
60 SPEED = 255
70 HTAB C + N
80 PRINT "  ";
90 NEXT N
```

Line 50 prints the arrow. Line 80 erases the arrow by over-printing with blank spaces. If the arrow takes up two spaces , it is erased by over-printeng with two blank spaces. Motion is achieved by using the FOR NEXT loop to change the column coordinate. This program moves the arrow from left to right. Show students that changing line 20 to (20 FOR N= 39 TO 1 STEP -1) will move the arrow from right to left.

The next logical step is to move that arrow up and down. Introduce VTAB and back away. By now the students should "own the problem". Allow them some time to explore the dimensions of the text screen and to become comfortable with increasing and decreasing loops. Some smart kid might even discover that increasing the step to 4 or 5 will cause the arrow to hop,skip or jump. Using FOR NEXT loops to create motion allows us to visualize the construct of a loop in a very tangible way.

The "print - erase - move - print" mehtod will allow right , left , up , and down motion , but is nevertheless limited . In text mode it is limited to an image that can be achieved using text characters. The image can be a word (try the program above with your name in line 50) , a symbol , a short sentence, or a combination of the above. The image cannot be Pac Man.

Pac Man requires the use of graphics mode. Use of graphics entails learning some new vocabulary. Fortunately the BASIC language closely parallels the English language. HLIN is easily remembered as a horizontal line , VLIN as a vertical line. PLOT is a command which many students understand because they have used it in math class. Familiarity with the coordinate system of the Apple screen can

be achieved in several ways. A " Name the Point " guessing game can be a great teaching tool. LoRes Apple Graph paper is indespensible. The variety of pictures that can be made by shading in blocks on graph paper is surprising. Kids have made pictures of everything from birds to trains , from dinosaurs to teddy bears. The possibilities are seemingly endless. If they can put the image on graph paper they can learn to put the picture on the computer screen The program below draws a truck.

```
5  Rem ** LoRes graphic picture
10 GR
20 COLOR = 9
30 HLIN 0,5 AT 20
40 HLIN 0,6 AT 21
50 HLIN 0,7 AT 22
60 PLOT 1,23
70 PLOT 5,23
```

Line 10 puts the Apple into graphics mode. Line 20 colors the truck orange. Encourage children to experiment with color. Let them discover that when COLOR = 0 the truck disappears. A bright child , familiar with the " print - erase - move - print" technique will try to make the same idea work in graphics mode. Instead of blotting out the image with blanks as is done in text mode, the student must erase the picture by drawing the image twice. In the program below the picture is first drawn in orange (COLOR = 9) and then drawn again in black (COLOR = 0). The two pictures must be exact duplicates, drawn at exactly the same coordinates. The second picture (COLOR = 0) deletes the first.

```
5 REM**This program illustrates
the "print - erase - move -
print" method of animation in
LoRes graphic mode.
10 HOME :GR
20 C = 0 : R = 0
30 COLOR = 9
50 GOSUB 100
60 COLOR = 0
70 GOSUB 100
75 C = C + 1
80 IF C = 30 THEN END
90 GOTO 30
100 HLIN C , C + 5 AT R
110 HLIN C , C + 6 AT R + 1
120 HLIN C , C + 7 AT R + 2
130 PLOT C + 1 , R + 3
140 PLOT C + 4 , R + 3
150 RETURN
```

Animation is more than drawing and erasing. There must be motion. To simulate motion, a technique for changing the coordinates of the point being plotted is needed. This is most easily done by naming the point with variables (PLOT C , R rather than PLOT 10 , 20). A variable can be incremented (LET C = C + 1) causing the image to move. Line 20 tells the computer to begin drawing at column 0 , row 20. Line 75 moves the drawing one column to the right. Line 80 stops the truck before it drives off the screen. If you wish perpetual motion change line 80 (80 IF C = 30 THEN 20). This program has an adaptable skeleton ; it can be easily modified to give the appearance of motion to any picture which can be drawn by shading blocks on Apple graph paper. It is possible to change the picture being animated without rewriting the entire program ; only the lines with line numbers greater than 90 need be replaced. The image , as it moves across the screen ,flickers and flashes. The smaller the picture, the smoother the animation. The

larger the picture , the more flicker will be observed.

Bright students will notice that C = C +1 moves the picture left to right. Given time to experiment , modify and "mess around " the student will discover that C = C - 1 moves the image right to left , R = R +1 moves the picture down and R = R - 1 moves the picture up. Changing both column and row coordinates at the same time give the appearance of diagonal motion.

The program can be easily modified to animate a high resolution image. Keep in mind that the low resolution screen has columns 0 to 39 and rows 0 to 39 (1600 points can be plotted). The high resolution screen has columns 0 to 279 and rows 0 to 159 (44800 points can be plotted). Try the program below.

```
5  REM** This program
illustrates the "print - erase
- move - print" mehtod of
asnimation in HiRes graphics
mode
10 HOME : HGR
20 C = 0 : R = 20
30 HCOLOR = 3
40 GOSUB 100
50 HCOLOR = 0
60 GOSUB 100
70 C = C + 1
80 IF C = 270 THEN END
90 GOTO 30
100 HPLOT C , R TO C + 5 , R
110 HPLOT C , R + 1 TO C + 6 ,
R + 1
120 HPLOT C , R + 2 TO C + 7 ,
R + 2
130 HPLOT C + 1 , R + 3
140 HPLOT C + 4 , R + 3
150 RETURN
```

The truck is smaller; the motion is slower. The truck can easily be made larger by changing the lines with line numbers greater than 90. Speed can be increased by changing line 70. When C = C + 1 the truck is a tortoise. Change line 70 to 70 C = C + 10 and the truck is a hare. You can adjust line 70 to change a truck dawdling along to a truck whizzing past. Motion on the computer monitor is an illusion . The human eye is easily fooled.

Animations are fun for children and a never ending source of surprise for the teacher. Boys and girls have animated everything from a football game to the passage of food through the human digestive system. Time is appropriately spent on animations ; students learn programming because they must in order to make the picture move. Use of graphics encourages creativity. The end product is unique to the individual. Schools offer too many opportunities for children to imitate and not enough opportunities for boys and girls to create. Too often students learn BASIC by writing programs that make tables or do repetitive computations. These things are necessary and important , but do not generate the enthusiasm generated by animations. Many programming skills can be most easily taught through the use of graphics. Student motivation is increased , teaching is easier and learning is more fun.

A TEACHER TECHNOLOGIST TRAINING PROGRAM

Nancy Kimbrough
Department of Technology
5300 San Felipe
Houston, Texas 77056

The implementation of a computer literacy course in the Houston Independent School District necessitated qualified personnel as instructors. To meet this need, a position entitled Teacher Technologist was created. These teachers attend 296 hours of training in a variety of computer-related topics and instructional applications.

Teaching a computer literacy course requires a broad knowledge of computers and their applications in the classroom. Since few universities are offering courses to meet these demands, the Houston Independent School District has taken this responsibility upon itself, and designed a course of study for its teachers who will be teaching computer literacy. This course of study consists of 296 hours of training, with 30 hours of update training each year. The courses are offered in the evenings, on weekends, and during the summers, Some of the hours are self-study hours that involve classroom applications of concepts discussed.

Houston's Teacher Technologist training modules are grouped into five categories: Instructional Applications of the Computer, Computer Literacy, The Computer as a Tool, Computer Languages, and Planning and Implementation. Each of these categories is then subdivided into four-hour modules and self-study units.

An orientation session is held prior to the start of the training to familiarize the technologists with the training sessions, available pre-tests, dates and times for the sessions, and a question and answer period. The teachers are given the opportunity to meet with one another as they need to know the backgrounds and goals of the people with whom they will be spending 296 hours.

A pre-testing session of four hours is held to allow teachers to test out of selected modules for which they can demonstrate competence. Pre-tests are administered for the following modules:

Hands-on Training
Math Applications
Reading Applications
Hardware
Data Management
Electronic Spreadsheet
Word Processing

Pre-tests are also administered at a later date for computer languages. Teachers are not required to take the pre-tests nor are they penalized if they do not pass a pre-test. They may

also choose to attend a session for which they have successfully completed a pre-test.

One of the first courses in the training sequence is a four-module (16 hour) session entitled Hands-on Training. The sixteen hours of training covers such topics as: booting the computer, assembly, and disassembly of the computer, hardware and software terminology, special function keys, systems commands -- CATALOG, LIST, RUN, LOAD -- initialization of diskettes, common error messages, diskette maintenance, types of files, direct execution commands, procedures for terminating a program, DOS 3.2 and 3.3, diskette copying, instructional strategies, turnkey and conventional systems, and instructional applications.

A four-hour module entitled Tutor, Tool, Tutee is based on Robert Taylor's book, "Computers in Education: Tutor, Tool, Tutee". The philosophies of various experts in educational computing are presented and discussed. Examples of the three different modes of computer usage are presented, as well as software resources for each usage. The teachers are given time to preview some demonstration diskettes for Computer Assisted Instruction, to go through the Bank Street Writer tutorial, and to do some very simple BASIC programming.

Math Applications, two four-hour modules, gives technologists an overview of the K-12 math courseware being used in the district. Examples of various management systems are presented, and student records are created. Classroom application strategies are discussed, as well as scheduling strategies. Implementation into the existing courseware is also discussed. Teachers who are currently using some of the courseware packages are invited to share some of their experiences with them, including classroom management strategies.

A similar session is held for reading, and is entitled Reading Applications. It also consists of two four-hour modules. K-12 reading and language arts courseware are surveyed. Strategies for developing comprehension skills, improving sentence structure, and improving grammar skills are presented. Computer/student ratios are discussed, and suggestions for implementation for various ratios are presented.

One more session entitled Software Overview finishes the Instructional Applications sequence. It is also an eight hour session, consisting of two four-hour modules. This session gives the technologists an opportunity to survey other courseware packages on the district's approved courseware list. Evaluation techniques are discussed, and teachers are asked to complete a short evaluation instrument as they survey the courseware. Time is allotted at the end of the session to allow teachers to share their evaluations of the courseware they have seen and suggest methods for its implementation into the curriculum.

The first training session in Computer Literacy provides an overview of the history of computing and its societal implications. The four-hour session is entitled Computer Literacy: History and Society. The historical development of computers is presented through a filmstrip/tape series. Ideas for student projects in computer history are shared. Societal applications of computers are also discussed. Potential abuse and misuse of computers, as well as federal legislation concerning the right to privacy are topics for small group discussion. The potential impact of computer technology in education is also analyzed.

Another eight-hour session provides an overview of the district's K-12 literacy curriculum. The district's goals and objectives are discussed, as well as various strategies for implementing a computer literacy curriculum. Technologists are separated by grade level and given an opportunity to review commercially-available literacy programs and instructional materials. The computer literacy curriculum being used by each grade level is presented and classroom environments are discussed.

A four-hour session on hardware differentiates among the types of available hardware and applications for each. Technologists are asked to construct a matrix comparing costs and features of three different types of microcomputers and discuss their findings.

Available information resources are discussed in another four-hour module. Through means of a slide presentation, technologists are taken on a tour of the software library and shown many of the resources available for hardware and software information, as well as technical and instructional materials.

Twelve hours of self-study complete the literacy component; teachers are asked to incorporate the concepts discussed in the sessions into the classroom setting.

The third component of training presents the computer as a tool. Technologists learn how to use specific software packages related to business applications such as data management, electronic spreadsheets, authoring systems, and word processing. The above sessions are divided into four eight-hour sessions. An additional four-hour module discusses the various printers available and the capabilities of each. Special attention is given to the printers used in the district; technologists are shown various print fonts and print functions. Sixteen hours of self-study accompany this component in which technologists demonstrate specific applications of the software packages. Instructional needs and applications are considered, and project assignments reflect these needs.

In the fourth category, languages, technologists are asked to choose two of three languages offered. It is suggested that elementary teachers take Logo and BASIC and secondary teachers generally take BASIC and Logo or PASCAL. Each language is studied in five eight-hour modules with an additional eight hours of unit development. In addition to language concepts, teaching strategies and curriculum implementation techniques are explored. In an additional four-hour self-study module, technologists are asked to suggest strategies for incorporating computing languages into their subject areas.

The last category, Planning and Implementationm focuses on the goals and objectives of the school district and the technologists role in meeting these goals. In a session on planning and decision making, funding options, budget expenditures, hardware configurations, software requirements, and environmental considerations are presented. A variety of classroom settings and student scheduling options are surveyed in another session. Technologists are asked to design physical layouts for classrooms and labs given varying student/computer ratios, furniture, and space considerations. Software evaluation procedures are discussed in a sixteen hour session. Technologists are

also asked to complete a software evaluation form using a district-approved instrument. A four-hour session on hardware troubleshooting presents some hardware and software malfunctions and methods used by maintenance technicians to correct such malfunctions. The staff of technicians is introduced and the technologists are encouraged to use the technicians' expertise for maintenance of district computers. Two other sessions encourage technologists to keep abreast of emerging technologies and be aware of educational trends that can affect the future of computers in education. Upon completion of the 296 hours of training, the technologists realize this only marks the beginning of the learning needed to teach computer literacy in a world of rapidly changing technology.

Sports Related Microcomputer Use with Emphasis on Football Scouting and Analysis

Don Knezek
Box 745
Olney, TX
76374

This paper cites specific examples to demonstrate the potential of the microcomputer as a tool for use in an athletic setting, and offers guidelines for merging microcomputers into existing systems.

There exists great potential for microcomputer application in the area of athletics. Tasks such as individualized theory and technique instruction, budget planning and accounting, staff and student/athlete record keeping, inventory and purchasing, correspondence and report production, mailing and phone lists management, scheduling of meetings and facilities, and time and task management are being very successfully handled with the help of microcomputers in a variety of business and institutional environments. Individuals responsible for the management of athletic teams and programs can do well to look to these working examples as patterns. Other tasks, however, are more unique and require original techniques developed by those of us involved in athletics. Some examples of these creative opportunities are: team scouting and analysis of both self and opponent; predicted and actual performance analysis for the individual athlete; technique analysis; out of season development tracking; contest orgaization and management; competition simulations; recruit and booster contact; storage and retrieval of recruit information; team transportation arrangements; ticket and gate-pass management; and athlete treatment and rehabilitation records. The list goes on and is actually only limited by our imaginations.

In this paper the author characterizes examples of both commonly encountered applications and uniquely sports-related ones. Microcomputer based football scouting and analysis is used to demonstrate a detailed application. Guidelines are offered, too, for incorporating microcomputers into the athletic program. These examples and guidelines are based on the shared experiences of many persons involved with both microcomputers and athletics. Most heavily, however, they arise from the personal experience of the author while developing a microcomputer based athletic information system at Olney High School, and assisting in the development of such systems at other instituions.

As evidenced by the earlier sampling of tasks related to athletics, sports programs require consistent management of large bodies of infromation at practically every level. Enormous amounts of a staff's time and energy are devoted to manipulating information, be it in the office of the athletic director, academic counselor, business manager, coach, sports information director, recruiting coordinator, trainer, equipment manager, or transportation director. As athletic programs become more accountable for their diminishing slice of budgeted funds, improved efficiency in handling this information is a necessity for maintaining effectiveness.

There is a tool, now practically available to any athletic program, which is being used to help meet this challenge. This tool, of course, is the microcomputer. Let us first characterize a reasonable microcomputer system for athletic use and follow with some specific examples of how it can be applied.

A microcomputer system, as far as we are concerned, consists of three main components: hardware; software; and the operator (user). The hardware (physical equipment) includes a keyboard for entering information; a video screen for monitoring this input and for displaying instructions to the user; a disk drive or two for storing mounds of information permanently on floppy disks for future access; and a printer to furnish the user with printed documents based on information entered. These devices are necessary to make the computer an effective tool in our applications, but detailed technical knowledge of them is not critical. Other items, such as the computer (processor) itself and devices for linking computers together (networks and modems), are necessarily present but are not handled directly by the user and are therefore omitted from our list.

Before this impressive collection of hardware can do any work at all for us it must be driven by a set of very detailed instructions called a program. These programs (normally) are magnetically coded on floppy diskettes and remain invisible to the user. Selection and use of a particular program is what determines, at a given time, whether the computer will fill the roll of a wordprocessor, a basketball statistician, a football analyst, a time and task manager, or any of thousands of other possibilities. The program itself is only half of a software package, however. The set of printed instructions (documentation), which accompany the program and enable the user to operate the program properly, is the other element making up this part of the system.

The human operator, or user, of the microcomputer is the final component. This person, who actually enters and extracts information and directs the overall operation, rounds out our system.

Probably the most widespread use of the microcomputer is word processing. From professional writing to casual correspondence, typewriters and yellow pads are giving way. Athletic handbooks, playbooks, memorandums, letters, schedules, reports, rosters, itineraries and press releases are being produced, modified, stored and retrieved with greater efficiency using the microcomputer. As a specific example, announcements and invitations for Olney High School's 1984 invitational track and field meet were personalized, printed, addressed and mailed in a matter of two hours by a single coach. Using a microcomputer based word processing package the 1983 documents and address list were retrieved and updated in minutes. By automatically merging the address list and the form documents, the personalized invitations were produced. These were (manually) placed into school envelopes to which microcomputer produced mailing labels were affixed. Presto! A dozen man hours saved. The same technique will be applied in recruiting volunteer workers for the relays. The implications for booster contact and collegiate recruiting are obvious.

An application area more unique to athletics is that of basketball statistics management. A good selection of reasonably priced software is available for this task. Simply, the microcomputer accepts raw information normally gathered during a game and produces comprehensive statistical reports on both individual and team performances for a single game and/or over several games. Microcomputers at OHS are applied in this way by an assistant girls basketball coach who had never used a computer before the '83-'84 school year. These statistics are available shortly after completion of a game and enhance video tape analysis and practice planning. Too, as the season progresses, developing trends are evident. Most teams and individuals in an athletic program can benefit from similar analysis.

The most useful single piece of software an athletic department can have access to is an information management package. This application enables an operator to use the microcomputer to store, organize, retrieve, sort, modify and report infromation on any subject imaginable. Some common uses include staff and athlete personal profiles, inventory records, mailing and phone list management, recruit information storage and retrieval, scheduling and time management, booster infromation and goal/performance records. At Olney High School we have a hall of fame honor award. Selection is based on a fairly elaborate point scale which rewards both participation and achievement through all four years of high school. Records for each of our

student-athletes are kept such a data base to help guarantee accuracy and continuity, especially in case of staff turnover. A candidates hall of fame status can be easily retrieved at any time and obscure rosters can be reconstructed in seconds. Software in this category is plentiful, but varies greatly in capacities, ease of use, speed of retrieval and report design capabilities. Most, however, does allow the user flexibility in the design of input format.

Budget and accounting records are also easily maintained with the help of the microcomputer. Packages for this application vary as much in complexity, power and price as in any area. The author uses a popular home accounting package to manage his track and field budget quite successfully. This is becoming a quite common application in the athletic community.

But of all microcomputer uses in athletics the one nearest and dearest to the heart of the author is that of football scouting and analysis. Coaches responsible for this duty devote enormous amounts of time and energy to production of accurate and comprehensive statistical reports both on opponents and on themselves. Pressures of producing these reports quickly and from an information-intensive base of data often result in neglect of "normal" educational and personal duties. Too often this overload results in excellent educators leaving the profession.

With 20 or so easily identifiable parameters per scrimmage down, and a minimum of 80 plays per game, thorough analysis of even a single game can be a formidable task. Multiply that by an average of five games of opponent analysis and five games of self-analysis and you have outlined a fair week's worth of football scouting. And we have not even considered analysis of previous season, special situations, and game in progress.

Though collection of this data and interpretation of results still must be done by skilled staff, the tedious sorting, computation, organization and presentation of results is being successfully handled by microcomputer. We at OHS have been through two full seasons of automated self and opponent analysis, including both during game and postgame/pregame analysis. We find the system results in earlier, better and more clearly organized reports. Additionally, there are two advantages over mainframe based scouting systems: control of the system is within the department rather than at some far-removed computing center, and work locations and times are extremely flexible with the relative portability of the micros.

An average of 45 minutes is required to enter infromation from each game, and more information is reported than can possibly be put to use. But input duties can be delegated and the ability to select useful information from the mounds made available improves with experience.

So exactly what is the holdup on moving to a microcomputer based system? Until recently ther have been three stumbling blocks. One, however, the lack of microcomputer hardware, has been removed by drastically lower prices and the appearance of microcomputers in academic and vocational programs in our schools. Many micros in our schools remain idle during evenings and weekends when coaches need them most. The other two, however, remain: lack of qualified staff members who feel comfortable with microcomputers, and lack of quality software to do the job.

The following guidelines, though aimed primarily at automated football scouting systems, apply with few exceptions to any athletic application of microcomputers.

First heed a few words of caution. Do not abandon a system that is now working well to dive head first into microcomputer scouting. At first, use both systems simultaneously and use one to verify the other. Use restraint in allocating personnel, money and primary responsibility to this new system and approach it with realistic expectations as to both efficiency improvement and effetiveness of results acquired. Most importantly understand that there is going to be a transition period as you become accustomed to the required exactness of the computer and to the idea of often receiving more information than you can use. This is normal and is only temporary.

There are some key considerations to keep in mind to help make the transition to microcomputers as smooth and inexpensive as

possible. First, use hardware which is already available to you. Any of the reputable microcomputer brands will be more than adequate if the software is available for it. Train your personnel in one of the computer literacy courses now available. Not only will the training aid in the use of microcomputers in scouting, but it will enhance general credentials. Seminars are available for specific athletic applications once a basic foundation is obtained.

Most importantly, purchase an inexpensive ($250 or less) software package which is well documented, which comes recommended, and which runs on the microcomputer you have available. Do not pass up a program or package because it does not do everything you want it to do. Even if you spend $2000 on your first program, it is almost certain to prove unsatisfactory. This need not be the case for an application in which software development has progressed beyond infancy (word processing, for example). Your first computer scouting and analysis package should teach you the fundamentals soundly. Not until you have finished a season on the computer will you have the experience to shop effectively for a "permanent" software package. Time your purchase during the offseason so that you can become familiar with the system before the season. Again, be careful not to spend enormous resources on a package that you will almost surely discard.

After your transition season, you will go on an intelligent search for that software (and possibly hardware) package which can give you all the benefits you saw glimpses of during your first exposure. You will know what features you do and do not want. And though software is improving much more slowly than hardware, it is improving. There may be a better package for your system on the market by next season.

There is help and advice available to coaches concerning implementation of microcomputers in athletics. The following may prove to be NOT such good sources: magazine/catalog advertisements, brochures/ pamphlets, clinic exhibits, computer/software vendors, and professional programmers. You would not go to a sporting goods salesman or sports reporter to learn to run the veer offense. You would locate other coaches who are running the veer successfully and seek their help and advice. Similarly with microcomputer applications, find someone in your position who is doing what you want to do, and is having success with it. Study what he or she is doing and modify it to fit into your scheme. We are fortunate so far in athletics in that ideas and experiences are shared without the ugly consulting fees that often accompany expertise in other professions. Enjoy it while it lasts; the microcomputer in athletics is on the brink of becoming big business.

Alternative High School Use of Microcomputers

Robert Lee, Jr.
Valley View Learning Center
Carrollton-Farmers Branch I.S.D.
2582 Valley View Lane
Farmers Branch, Texas 75234

The objective of this paper presentation is to inform secondary school teachers and administrators of the uses being made of microcomputers in a small alternative high school serving potential and former dropouts. A brief description is given of the specific objectives of the computer applications, the computer programs used, and the application's successes and failures. Applications to be described include word processing for attendance counseling, test review for physical science, and paragraph writing for English.

Introduction

Computers are beginning to have a prominent place in the Valley View Learning Center alternative school. Applications for the machines are beginning to make themselves more apparent. No longer are we inmpressed with the computer's ability to provide drill and practice. The computer is being used now as a tool at VVLC to provide assistance in improving attendance, reviewing for exams, and developing paragraph writing. With the school's focus on individualization of learning, the computer has the potential to provide each student with what he/she needs when it is needed. The problem comes in finding the applications that can be handled by the available machines and software.

Valley View Learning Center is an alternative high school within the Carrolton-Farmers Branch Independent School District. It is designed to provide an opportunity for students that have dropped out of school and want to return to school without the pressures of a traditional high school. VVLC also enrolls students not able to function satisfactorily in a traditional high school. Such students and their parents may request enrollment at VVLC but because of the school's limited size, enrollment is a priviledge.

Ten teachers, counselor, secretary, and building principal serve a student population of 150 students ranging in age of 15-20 with a waiting list of 5-10 students. The traditional subject areas are offered (English, math, social studies, science, P.E.) in addition to reading, art, and typing as well as two work study programs.

Students are divided into morning and afternoon sessions; those in the morning must have a regular job and tend to be older students. There are as few rules as possible governing student behavior. Regular attendance is expected but allowances are made for personal business and sickness. There is no schedule of classes (no bells!) but students are enrolled in specific courses. Students attend the classes in which they are enrolled at any time during their assigned session and stay for as along as they want or need. When an assignment is complete, the student presents a slip to the teacher for a signature of

attendance/performance and leaves the classroom for another subject. When a student has attended all his/her classes, the student leaves the attendance slip with his/her advisor and leaves school for the day. Because students are in different materials, at different places within different courses, and present in class at different times, it is near impossible to make group presentations. The teacher's role thus changes from one of disseminator to one of facilitator.

The three program applications discussed are just a few of the computer programs being used at VVLC. They represent though some of the more interesting computer applications in our school; word processing in an attendance program, test review in physical science, and paragraph writing in English.

Attendance, Counseling, and Word Processing

For the student population our school serves, attendance is a primary concern and many student's primary problem. One method used this year to encourage better attendance is a system of points given for attendance and for lessons completed each day. The counselor collects the student's accumulated points total each week, and if satisfactory, generates a letter to the parent. The letter describes the student's attendance and class performance, and that the student can use the points to 'purchase' school t-shirts, tote-bags, paper weights, or soft drinks. To be redeemed for the items, the letter must be signed by the parent and returned by the student.

As many as 50 letters must be generated each week and without a word processor the task would be too great. To fascillitate the letter writing, the counselor generates a form letter on the computer leaving codes in place of the student name, date, and number of points. A file of student names, date, and points is then created. The word processing program then merges one student's name, date, and points into the form letter replacing the codes. The letter is then printed by the word processor. When complete, the next set of data is enterred by the program and the next letter is printed. This process is repeated until all the data in the file has been used and all the letters printed. Each letter is then signed by the principal and sent home with the student for a parent's signature.

When first implemented, the letters were mailed home but we found there was no guarantee the parent would pass the letters on to the child. Also, the time-lag in mailing as well as the mailing cost became a problem. By giving students the letters, an opportunity existed to further reinforce the student's performance as well as providing a backup system in making sure all qualified students received their letters. Generally, if a student did not receive a letter and felt he/she should have, they would complain to their advisor, to the counselor or to the principal.

The system seems to be working. Parents have expressed interest and approval in hearing something positive from school and appreciate the effort the school is making in helping their child. Many students are interested in their points and compete for the highest total. Some students use their points immediately while others collect them for large purchases while still others collect letters but seemingly, will never redeem them. Still others do not see the system as important to them and continue to have poor attendance and class performance.

The problems in the system are few. The couple of hours a week required to collect the points and generate the letters does not seem excessive in light of the benefits. Most of the

time is spent nursing the printer into doing its job properly. Character qualitiy of the dot-matrix printer could be better but no one has complained or even mentioned it. Also, the fact that the letters are computer generated has had no negative effect on the students or their parents.

The 'token economy' has been in effect since September, 1983 and continues to be seen as important by a large group of students and their parents. The use of 'letters' as the means of recording and communicating to the student and parent makes the school's objectives clear to everyone. Generation of such a large number of letters each week would be impossible without the use of the word processing computer.

Test Review Programs in Physical Science

The Test Review Programs are designed to help students prepare for objective exams by presenting questions similar to those the student will see on the exam as well as help when an incorrect choice is made. Each review attempts to minimize a student's surprise and confusion by providing the student with questions similar in content, format, and difficulty to those the student will see on the exam. When a student answer is wrong, the computer responds with an explanation why the answer is incorrect and then provides another opportunity to answer the question correctly.

The key to the programs is the help the computer provides when an incorrect choice is made. Students seldom have an opportunity to understand test questions until after the exam when the class may 'go over' the exam with the teacher. The computer can 'go over' each objective before an exam by presenting questions similar to those on the exam. If an incorrect choice is made, the computer can respond with what the programmer feels is the reason why an incorrect answer was chosen.

Other methods can provide a review of the test material but only the computer can demand interaction from each student user and provide immediate student feedback. In addition, since the computer responds with why a particular choice is incorrect, the computer can individualize each review to each student. The student need spend time only on questions with which she/he is having trouble. If a mini-tutorial is necessary for understanding, the computer can provide it at the moment the student missed the question. Since the computer is infinitely patient, the student can review as long as necessary on the machine and repeat the review as often as each student feels is required.

Each Test Review Program is ten to twenty questions. The questions are presented individually in the formats to be seen on the exam: multiple choice, multi-answer multiple choice, matching, and/or true-false. Illustrations, tables and/or graphs are included also if graphics are used on the exam and the test objectives require them.

A summary of how well the student did on the review follows the last question. It lists how many questions were answered correctly on the first try, how many on the second try, and how many on the third try. The student decides then if she/he is ready for the exam or if more review is necessary.

In the three years I have been using the Test Review Programs, most of my students have responded favorably towards the computer reviews. Adequate review for exams is not normal behavior for alternative school students. They go through the motions of learning the material and if they don't pass the exam, they're not surprised - they feel they are not

smart enough and the teacher didn't teach it to them anyway. Some students do not like the machines and prefer not taking the computer reviews. Some students become upset and appear almost lost without the computer reviews. The review seems to be a support in their mental preparation for the exam. If it's not there, they don't feel nearly as confident as when they do the computer review.

The Test Review Programs give students an opportunity to see the type of questions they'll see on the test, receive feedback when their answers are wrong, and a final opportunity to evaluate their own knowledge of the subject. Although the computer reviews are used in physical science, any class that uses objective exams can benefit. The problem is that there are few programs available that present students questions with feedback on wrong answers. To create such a program, besides creating the program itself, generating questions and responses to wrong answers require a lot of teacher time. I feel such programs would be an appropriate addition to a textbook publisher's catalogue.

Paragraph Program in English

The idea for the 'paragraph program' came out of a need to get students to express themsleves in a paragraph rather than in a simple sentence. Many students have trouble getting past the first sentence. The initial thought is there but the supporting thoughts don't come so easily. The program provides the student with prompts in writing a complete, well developed paragraph.

The program begins with a menu of topic sentences. The student chooses one by number. The screen clears and the topic sentence is written across the screen. In a basic expository paragraph, a transition word such as 'because' follows the topic sentence. Prompted by the topic sentence and the transition word, the user enters a sentence. Upon entry of the sentence and followed by RETURN, the computer prints the next transition word, 'for example' and waits for a response. The student completes the thought, pushes RETURN and 'in addition' follows. Again the student enters a sentence and pushes RETURN. The topic sentence is now repeated followed by 'because'. The student completes the sentence and the computer adds 'also'. Another student sentence is enterred and followed then by 'in conclusion' and the student completes the paragraph.

A menu offering choices of printing the paragraph to the screen or to the printer is then given. Whichever choice is made, the paragraph is printed in paragraph form as enterred except for the deletion of the transition words.

The strength of the program is that it forces the user through a paragraph mold. The computer is infinitely patient but demands some kind of response. Whatever is enterred is then reprinted in a new form. The user will then reread what he/she has written with a more critical perspective - something students seldom do when writing in long hand.

The form demanded by the computer's use of topic sentence and transition words demonstrates the writing of a basic paragraph by directly involving the student in following a paragraph 'formula'. By changing the topic sentence, number and type of transition words, different types of paragraphs can be generated by the student such as paragraphs of process and compare-contrast.

Beyond teaching the paragraph, the paragraph program has applications in guiding the student through the writing of short papers. If a brief outline were first written, Roman Numeral I would be topic sentence

'one', Roman Numeral II would be topic sentence 'two', and so on. The paragraphs would be linked together by the logic of the outline and each topic sentence would be expanded upon within each paragraph.

In the few months the program has been used, the more advanced students have trouble using the program; they feel too restricted in the formula it provides. But as a general rule,most students seem to enjoy and benefit from the program. Besides providing these students a form to follow in composing a paragraph, the program encourages students to better edit their writing. The ease with which one can read a typewritten copy seems to motivate us all to work at improving our written communication.

Conclusion

For now, computers are limited in what they can do in the classroom. Those limitations may decrease in the future but for now, the computer can perform only a few functions for the student and teacher. It seems reasonable to expect that the computer can play a more important role in a student's education. As educators, it is part of our job to expand that role and help our students become a better master of the machine. The three applications described play an important part at VVLC never really dreamed of three years ago. I'm looking forward to the next three years.

Scott, Foresman and Company 1900 East Lake Avenue Glenview, Illinois 60025 312/729-3000

Presentation Title: Computer Literacy: An Integrated Approach

George A. Logue
Scott, Foresman and Company
1900 E. Lake Avenue
Glenview, IL 60025

Abstract: This session is a presentation of a comprehensive computer literacy program, the Scott, Foresman Computer Literacy Courseware Series. The program is designed for use with Apple microcomputers and is appropriate for intermediate through high school students. Special attention will be devoted to demonstrating how computer literacy instruction can be integrated with various curriculum areas.

The following essay has been prepared by Don G. Rawitsch an author of the Scott, Foresman Computer Literacy Courseware Series. Mr. Rawitsch is Manager of User Services for the Instructional Services Division of the Minnesota Educational Computing Consortium in St. Paul, Minnesota. In his essay, Mr. Rawitsch succinctly defines computer literacy in terms of student outcomes. The presentation of the Scott, Foresman Computer Literacy Courseware Series will detail this definition of computer literacy. The demonstration will show how educators in all curriculum areas can become successful computer literacy instructors.

"Take a walk through your town or city. You won't go more than a block without seeing some evidence of computers at work. The Industrial Age gave us numerous developments that reshaped the world--the steam engine, electric light, automobile, and airplane, to name a few. But the invention and widespread use of computers has ushered in a new Electronic Age. It is the age today's students live in--the age they must learn to understand.

The Need for Computer Literacy

History has shown that the key to power in a society is the control of information. Computers provide us with a powerful tool for doing this. The control of computers and related information technology will therefore play an important role in the way power is distributed in our society. For this reason, it is critical that the citizens of today and tomorrow obtain fundamental knowledge and skills about this technology. The average citizen need not become a computer expert. However, he or she does need to become "computer literate."

What is computer literacy? There are perhaps as many definitions as there are people attempting to define it. In general, computer literacy is an understanding of computer-related topics that will allow a citizen to function comfortably in a computerized world.

A more detailed definition, of

relevance to educators, should consider student outcomes. In the case of computer literacy, we are looking for outcomes that will help students face a technological society. Following is a list of five topics that embody the author's definition of computer literacy. A justification for each is given in te-ms of specific student outcomes.

1. *How Computers Are Used*--we want students to understand that computer use is all around us; students should also know what kinds of tasks computers can do;

2. *How Computers Work*--we want students to overcome the feeling that technology is a mystery only experts can understand;

3. *How Computers Are Operated*--we want students to be self-sufficient in the use of computers so they can use them for their own purposes;

4. *How Computers Are Programmed*--we want students to understand that people control computers; learning a little about programming will help accomplish this;

5. *How Computers Impact Society*--we want students to understand that computer use brings both benefits and problems, and to understand how those problems might be solved.

If students attain a reasonable understanding in each of these areas, they may be considered computer literate.

Teaching Computer Literacy

Computers are in frequent use today to teach subjects such as reading, mathematics, science, social studies, and foreign languages. Ironically, computer literacy is itself often taught with only a textbook. This is a mistake. Students should use computers to learn about computers.

In addition, computer literacy need not be confined to the mathematics curriculum, but can be included in various curriculum areas.

For example, Scott, Foresman Computer Literacy Courseware Series fits well into the social studies curriculum. Most of the lessons cover ways computers are used in society and how they impact society. Many of the lessons raise issues which have no clear-cut answers. Students are encouraged to express their own opinions in addition to acquiring specific content knowledge.

Scott, Foresman Computer Literacy Courseware Series offers enough flexibility to make it useful in other subject areas as well:

Reading--computers motivate students. Thus, students will have an intrisic interest in reading the material. Also, the books contain many new words which will enrich students' vocabularies.

Mathematics--some of the example computer applications involve simple mathematics, especially as related to real-life situations. Students who have used the computer in mathematics class will find that the books reinforce their work.

Business--many of the example computer applications relate directly to the world

of business. The books can thus be used to make business education material come alive for students.

Computer Science--the schools that stress teaching about computers as machines will find that the books add a new non-technical perspective to the subjects of computer operation and computer programming.

The development of computer literacy should be an important objective of education today. We, the educators, need to provide students with a wide range of computer experiences in a variety of subject areas and grade levels. In so doing, we can better prepare today's students for life in the computerized world in which they live."

AUTOMATING A LIBRARY MANAGEMENT SYSTEM

Susann Luper-Foy
7918 Jones-Maltsberger #J5
San Antonio, Texas 78216
(512) 824-0046

If you decide to automate your library, how should you select the system to use? In this paper, I suggest the following main guideline: the new system should perform the same types of tasks as the old, manual system, only better. Hence the main problem you encounter in selecting a new system is describing the functions performed by the old one. It is the goal of this paper to provide an outline that will simplify this task for those with little expertise in computer science.

A common approach to choosing a computer system is to simply buy the first affordable system encountered and hope for the best. In implementing it, you will learn first-hand exactly what its drawbacks are and what special operations you would really like to have in such a program. Although in the end you may not have the optimal system, you will have spent the least possible amount of time in the selection. Moreover, you will most likely be more satisfied with this system than with your previous manual library system since nearly any automated system will have advantages over its manual counterpart.

The potential drawbacks of this approach are quite obvious. There is risk involved in not fully understanding what it is you need and what exactly you are purchasing. For example, you may find in retrospect that you were overly impressed by a fancy sorting routine which you rarely find use for while you overlooked the need for a simple feature that would have circumvented frequent typing errors by library aides. Also, salespeople are of very little value as an objective source of information since most of them know nothing of the alternative software, and it is in their interest to promote their product exclusively.

There is a better way to select your software. Think of the library as an information processing system; various types of information enter the system and are appropriately processed. Loan requests, for example, are granted or denied, purchase orders are paid, overdue fines collected and recorded, etc.

When the decision is made to automate an information processing system the objective is normally to process the same types of information, but to do so more efficiently. In the automated system, time consuming, repetitive tasks are directed to the quick, reliable machine in order to (1) free human time for more creative tasks, and/or (2) increase the volume of information processed by the entire system (in the case of the library, more patrons per hour/day, a larger collection of circulating books, etc. In any case, the goal is rarely to totally redesign the logic of the library system.

Once the task of selection is seen from this angle, very useful guidelines suggest themselves. The main point is this: the new system should process all of the same types of information as the manual one it is to replace, but more efficiently. In other words, it should perform all of the functions of the old system (maybe more), and with greater speed and accuracy. Therefore, major changes should occur in the mechanics but not the logic of the system: bar codes replace catalogue cards, a computer data file replaces sorted book cards filed in a desk drawer, and so on.

Another point suggested by the fact that the goal is to perform all of the old functions only better is this: in order to tell which automated system is best, you must first have a clear description of all of the various types of thing your old system does. These are the features to be looked for when evaluating software. The best software is the software that performs the old functions only better. Selection of hardware will depend almost entirely on the software chosen, and the limits of the total budget for the conversion project. Therefore attention will be focused almost entirely on describing software features. In sum, then, I propose the following method of selecting an automated system: (1) develop a clear formal description of your existing system, assigning priority rankings to various features, (2) evaluate available software according to how well it performs those tasks, and (3) decide on, purchase and implement a hardware/software configuration. By far the greatest portion of effort and attention will be required in the first of these three phases--the description process. If you are managing a public school library, it is unlikely that you will have funds to hire outside computer consultants to perform a systems analysis and/or program a customized library management system. It will therefore be up to you to outline the objectives of the automation project.

At this point I wish to provide tools for arriving at the information to be included in this formal description. I will present them in the form of a list of general questions to be answered by the librarian for a particular library system. The information provided here results from a project in which I developed a formal description (a system analysis) of an elementary school library. This report was to aid the school's administration in making a decision regarding automation of their library. Data for the report were provided by the school librarian, several software vendors, hardware vendors

(mostly price quotes), and other librarians[1] at various stages of the automation process.

I. Characteristics of the Library

(This is the information to have clearly outlined before approaching software vendors.)

1. volume

number of items in collection, separated into circulating and non-circulating

anticipated expansion over several years

number of patrons, including students, teachers & staff, and others

approximate number of transactions per day, per week

2. budget

total implementation budget plus any allowance for maintenance costs

separate limits for hardware and software?

(you may locate the ideal software package but to your dismay discover that it is only compatible with a hardware system $1000 beyond your total budget)

is there funding for temporary staff during the implementation period?

(for example, typists to enter titles into data base)

3. staff

number of hours of assistance per week

extent of computer background of library aides

4. library in relation to rest of school and school district

are there already computers in the building? in the library?

would any system you purchased be dedicated or would it be used for other administrative or educational activities, possibly requiring the hardware to be moved to classrooms for parts of the week/day?

are plans being made at the district level that will affect your library? (for example, a proposed network system involving all schools in the district)

5. details of routine procedures

list of files kept (shelf list, overdue books, classroom rosters, purchase orders, consideration list, etc.)

processing check-outs and check-ins

procedure for new acquisitions
issuing overdue/fine notices
processing each new book from consideration list to shelf
note any physical procedures that cannot be performed by any computer system such as reshelfing books, covering new books with protective plastic covers (this is to help in realistically assessing the amount of time saved by the automated system)
list procedures you would or would not be willing to sacrifice in order to adapt to an automated system
list procedures you are willing to continue to do manually, i.e. insignificant enough that they do not warrant purchasing a computer separate program for; for example purchasing magazine subscriptions, or recording lost books.
is there a need for security measures in your school?

6. job description of librarian
list any significant responsibilities not mentioned above; for example, blocks of time that you must be away from the library on a regular basis, or when you would be available to perfom computer updates, back-ups, etc.

7. other helpful information
physical layout of the library room
all facilities available for library use
especially list any idiosyncracies of this library not likely to be allowed for by software developers

II. Characteristics of the Software System
(This list along with the above is to help in sifting through the masses of software information provided by vendors.)

1. total cost
which hardware is it compatible with
include hidden costs, for example special paper products to match a particular printer

2. expected support from vendor
waranties on hardware and software
terms of possible maintenance contracts
promise for free software updates and replacement of worn disks

overall reputation/stability of company

3. documentation and training
   are manuals complete and appropriately written for your level?
   is there plenty of information on what to do in jam situations?
   free expert training for primary user

4. time considerations
   can entire package be delivered during school break to allow implementation?
   anticipated initial time investment for implementation
   computer response time for switching modes, processing check-outs, etc.

5. allowances for external (human) error
   does it perform automatic back-ups to minimize loss of data in the event of power outage, accidental bumping of ON/OFF switch, and other common mishaps?
   bar code entry vs. keyboard typing of names, call numbers, etc.
   protection during disk swapping: will the program recognize an oversight and
      - prompt for correction, or (at least)
      - refuse to continue on disasterous course
   protections from unauthorized access to data files (passwords)

6. general convenience
   does operation require frequent swapping of floppy disks?
   how frequent and how long are delays for changing modes?
   computer response time for sorts, searches, etc.

7. adaptability to details of your established procedures
   how much flexibility is allowed in formatting of information
   for example,
   -will it demand student addresses be entered even if this is not part of the normal library practice?
   -will it allow you to add comments to record of purchase?

8. adaptability to future developments
   expandability during growth in collection size
   will hardware be obsolete in two years? (this will affect your ability to get service as well as the amount of current educational software available, in case you do share the computer with classroom teachers)
   what are the limiting factors of the program?

maximum number of items in collection? maximum number of transactions processed per hour? per week?

9. general reactions of other librarians using this system
   disappointments
   suggested refinements
   regrets

The description that you will arrive at upon applying this outline will naturally be less formal and will use less technical language than would a professional systems analysis report, since it is primarily for your own benefit. However, it should be just as complete. The clearer your idea of the requirements of your library, the better able you will be to critically evaluate available software. Most of the required information regarding the library system is knowledge you already have, in the form of routine habits, which you need only organize and write in a formal, detailed style. Incidentally, it is always conceivable that you will decide as a result of your study that automation is not appropriate for your library this year.

---

[1]Special thanks to Gloria McClenahan and the Liberty Hill Elementary School for generous amounts of information and the chance to try out the computerized library management system being implemented in that school.

How We Did It: San Angelo's 1st Year - 6th Grade Computer Awareness Program

Jeff Miller
100 N. Magdalen Street
San Angelo, TX 76903

Steps in Implementing San Angelo's 1st year - 6th Grade computer program

By August of 1981 San Angelo Public Schools had a variety of computer equipment in place. Our Administrative offices, two high schools, three junior high schools and three elementary schools had various equipment, brand names and models. Others were anxious to buy. There were no guidelines for purchasing or use.

In September of 1982 Mr. Rex Lowe, our system's Assistant Superintendent-Administration & Instruction, formed a committee to recommend directions for computer use in our school district.

The committee was composed of people with knowledge of both computers and education: George Hill, Mathematics Supervisor; Wayne Breen, Computer Math Teacher; Richard Hooker, Director of Data Services San Angelo ISD; Sydna Easley, Principal Bowie Elementary School and this writer, Jeff Miller, Audio Visual Supervisor, comprised this committee.

The committee involved many other interested people including the business community and college level educators. The committee also worked closely with our Superintendent of Schools, Mr. Bill G. Graves and our Assistant Superintendent-Business and Operations, Mr. James E. Langford.

The committee soon agreed on an "Awareness Program" to be implemented at the sixth grade level, the contents of a curriculum guide and a plan for teacher training.

Hardware that would work with our over-all conception was investigated.

In the spring of 1983 Mr. Lowe took our work to the Superintendent who said it was ready for presentation to the School Board.

The Board Approved, the item was budgeted and, we let generalized bids for equipment types to local bidders.

Bids were accepted. Teachers were selected for Computer Assignments. Curriculum guides were written, printed and distributed. All equipment was set up for teacher training in July and August. Two computers with recorders were then set up in each of our schools with sixth grades.

At this writing our program is about a half a school year old. All our sixth graders have been exposed to a mini-history of computers, vocabulary drill in BASIC, Simple Flow-charting, Keyboarding and "Hands on" time at the computer.

A comprehensive evaluation will begin this spring. We now know that:

1. Most youngsters enjoy learning about and with computers.

2. A scheduled "Hands On" time is a motivating factor for many children.

3. CAI is a viable way to teach BASIC and any drill work.

4. More teacher preparation time is necessary.

5. A volunteer program, such as VIPS, is most beneficial.

6. More equipment is needed.

7. More supervision is needed.

8. More scheduling time is needed by schools, administration and teachers.

9. All students and all teachers are not necessarily interested in computers.

10. Keyboarding skills are essential. A course in "Keyboarding" prior to hands on computer work is desirable.

Our optimum plans would include training and equipment for "Computer Education" at the Junior and Senior High School level. All our futures computer-wise, however, will be affected by the flexibility and interpretation of H.B. 246.

SECURING TEACHER ACCEPTANCE OF TECHNOLOGY

Dr. Martha Muncrief
Consultant in Computer Education
Harris County Department of Education
6208 Irvington Boulevard
Houston, Texas 77022

ABSTRACT OF PRESENTATION

The Concerns-Based Adoption Model (CBAM) developed by Hall and Loucks will be examined and related to the use of computers in the classroom. A framework for staff development will be outlined with examples of appropriate activities at each level.

Introduction

Critics of education state with urgency their wish that teachers would be more sensitive, resourceful, literate, technologically-atuned, and committed to teaching. Having thus complained, the critics often fail to specify and/or provide resources for teachers to develop these useful qualities. In this paper we will deal with only one of these qualities....the ability and willingness to use technology in the classroom.

Historically there has been only minimal acceptance of technology and innovation in schools. Older techniques such as teacher explanation, chalkboards, and textbooks remain dominant in the classroom. Many reasons are suggested for teacher failure to use technology. There may be insufficient evidence of the effectiveness of the innovation in improving instruction. Computer technology is still immature with conflicting information regarding the value of the varied techniques for using computers in the classroom. Computers are costly and may utilize funds which teachers prefer to see spent in other ways. Furthermore, adequate educational software is still lacking in many content areas.

There is considerable evidence that teachers resist change for reasons which are more personal, yet often relate to their work as a professional. The teacher may perceive a high "cost" or "effort" involved in learning the use of a new technology....inadequate training has lent credence to this view. There may be a need to change teaching styles in order to incorporate new technology. The recommended practices may lack specificity....perhaps an administrative dictate to "do something with that computer in your classroom." Perhaps most important of all, many teachers view technology as a wedge between the teacher-pupil relationship. They fear losing the ability to work directly with the student in the learning process.

Concerns-Based Adoption Model

Among the recent studies dealing with teacher concerns, the Concerns-Based Adoption Model (CBAM) of Gene Hall and Susan Loucks provides a valuable format for examining the stages through which one might move in the adoption of an innovation. Initial assumption in the CBAM Model are that: (1) change is a process that takes time and is achieved in stages, (2) the individual must be the primary target, (3) change is highly personal, and

(4) stages of change involve both perceptions and feelings of individuals concerning the innovation as well as their skill in its use. It becomes apparent that staff developers need to diagnose their clients' location in the change process and assess the state of change as they adapt strategies along the way.

The chart below describes the Stages of Concern from the CBAM Model, a typical example of a teacher response at that stage, and some clarification of the stage. The intensity of concern at each stage will vary depending upon individual personality and pressure to adopt the innovation.

STAGES OF CONCERN

0 AWARENESS: I don't know anything about computers. (Little concern or involvement with the innovation.)

1 INFORMATION: I would like to learn about computers. (A general awareness of the innovation and casual interest in learning more.)

2 PERSONAL: I am concerned about how computers will affect me. (Uncertainty about the demands of the innovation and own adequacy in meeting these demands.)

3 MANAGEMENT: I am concerned about the time and resources involved in using computers. (Attention is placed on organizing and scheduling resources.)

4 CONSEQUENCES: I want to know if and how students are benefitting from the use of computers. (Focus is on the relevance of the innovation for students.)

5 COLLABORATION: I would like to share what we are doing with others. (Emphasis on coordination and cooperation in order to improve instruction.)

6 REFOCUSING: I would like to try other uses of computers or different types of equipment. (Exploration of more universal benefits from the innovation.)

Staff Development Model

The Stages of Concern provide a valuable format for the creation of a staff development model. It should be reemphasized that the introduction of an innovation in the classroom requires appropriate and extensive training of teachers. We know from the Rand studies that long-term training efforts are more likely to succeed than short-term ones. Administrators should therefore be prepared to commit time and resources to this effort over an extended period of time.

For convenience in implementing the model, we will group the Stages of Concern and describe some staff development activities which might be appropriate at each level.

LEVEL I: At this level we should attempt to help the teacher move from Awareness through Information to the Personal Stage of Concern. It is important for the teacher to realize the potential of the computer in the classroom. Extensive demonstrations should be provided of quality use of the computer in many modes: as a tutor to provide drill and practice, as an information-giving device, as a means of teaching problem-solving and higher-order thinking skills, as a means of manipulating words and data, and as a machine to be instructed. Teachers should be given opportunity to see computers used in all of these ways and time to "hash out" their philosophical objections and concerns about proposed changes.

Once a teacher has reached the Personal stage, there is likely to be a long lag time before moving further. Teachers need continuing opportunity for gaining new information about specific applications on a computer and opportunities for hands-on experiences. Since computer hardware can be overwhelming, it is extremely helpful to have an "expert" in the building who has the time, expertise, and assignment to assist other teachers in solving minor problems.

LEVEL II: In the Management stage the teacher begins to adapt the idea of a computer in the classroom to personal style and needs. Opportunities for further growth in programming skills and/or selection and use of courseware are essential. This is the time for administrators to affirm that adequate hardware, software and support for using computers will be available... or assistance in obtaining such.

Inservice training should focus on very specific application within a classroom. Small groups of teachers might be interested in "using data bases for teaching social studies" while another group may be concerned with "improving the teaching of language through the use of word processing." Math teachers might examine courseware which addresses the TABS objectives. Lower elementary teachers could examine activities with and without computers which help young children learn what computers do.

It might be noted at this point that most teachers will not move beyond Level II. They are already several steps ahead of the general population and expertise in computers is only one of many areas in which teachers are expected to be proficient. The teacher at this stage has already enhanced instruction in his/her classroom by using new technology more efficiently.

LEVEL III: Teachers at this stage need to be put in touch with others who are adapting the innovation to their particular situation. The creativity of these teachers needs to be shared in-house with appropriate rewards and recognition. This growing professionalism can be enhanced by gathering and printing unique applications of computers in the classroom. The CONSEQUENCE and COLLABORATION stages permit teachers to consult with each other as well as share with others.

Teachers at this level will continue to feel the need for information and personal growth. These needs are not likely to be met within the school's inservice format. Participation in professional conferences should be encouraged through released time and funding. The teacher may well want to increase competencies through university classes and/or a further degree. Whenever possible these teachers should be used as instructors and consultants to other teachers at a beginning stage of development.

LEVEL IV: The REFOCUSING stage becomes a catalyst for a deeper and broader level of awareness as the cycle begins again with AWARENESS at another higher and more complex level. Career and job changes may be indicated but the initiative at this stage probably rests with the individual. The work of the staff developers is complete....for the moment.

But have we evaded the big question? At what point has the teacher become "computer literate?" This fuzzy expression has different meanings in varying settings. If we assume that it means the ability to do what one needs and wants to be able to do with a computer, we might reasonably say that teachers at each of these stages are "computer literate." The Concerns-Based Adoption Model provides an insightful method of examining the personal ways in which we each adapt to innovations. Hopefully it will also provide a more helpful framework for the implementation of staff development programs for teachers.

REFERENCES

Berg, Roger. "Resisting Change: What the Literature Says About Computers in the Social Studies Classroom." Social Education, May 1983, pp 314-320.

Hall, Gene E. and Susan Loucks. "Teacher Concerns as a Basis for Facilitating and Personalizing Staff Development." Teachers College Record, Vol. 8, No. 1, September 1978, pp 36-53.

Mangieri, John N. and Richard E. Kemper. "Administrators: The Keys to Successful Inservice Programs." NASSP Bulletin, March 1983, pp 26-31.

McCarthy, Bernice. "Improving Staff Development Through CBAM and 4Mat." Educational Leadership, October 1982, pp 20-25.

TEACHING BASIS MATH SKILLS THROUGH COMPUTER CONCEPTS

LINDA NIX
Forest North, Round Rock ISD
Austin, Texas

This presentation is designed to acquaint the elementary teacher with computer related activities that help reinforce basic math skills of place value, computation, geometry, and problem solving. The abacus and Napier's Bones will be demonstrated. Student computer programs using LOGO and BASIC will also be shown.

The computer has become an ever increasing part of life in society today. Everywhere you find the computer has an influence. Children are aware of the computer in grocery stores, banks, airports, and even bowling alleys, so it seems only natural computers should be part of the elementary school curriculum too. Computers will be even more important in the immediate future as they become a part of every home and career. Through the study of some of the early computing devices children can see the development of computers. They will see how man is challenged to try new ways of doing things. Since the computer is based on mathematics, math class is an ideal setting to explore how the computer works, how it was developed, and how it can be a tool to help students learn basic mathematical and problem solving skills. This paper will show some concrete illustrations of how the computer and its ancestors can help elmentary school students develop a better understanding of some basic mathematics skills, such as place value, basic computation, geometry, and problem solving.

In my math classes, I have found place value to be one of the most difficult concepts for students to master. In an attempt to strengthen the understanding of the concept I have used many approaches and manipulative aids. When talking to children about the history of the computer and the many calculating devices which led to the development of the computer. I found the abacus to be an interesting aid to reinfroce the place value concept. The abacus is set up on the decimal system or base 10. As a child used the abacus, he could visually see the reason for grouping numbers. The first row of beads represents the numbers one to nine. When a child added another one it became ten and it was necessary to move a bead in the second row which represents tens. The numbers in the next row are hundreds, then comes thousands and so on. As children used a simple abacus which they had constructed they could do simple computations and vislualize the principle of place value at work.

Another early calculating device,

Napier's Bones, helps reinforce multiplication skills. This device was developed in Scotland in the 1500's by John Napier to make multiplication of numbers faster. The "bones" contain the multiplication facts in a specified order. When the "bones" are positioned or aligned in a certain way they produce the answer to a given problem. Some mathematics books refer to this method of multiplication as the lattice method. Using the "bones" is an interesting way to have children check their traditionally worked problems. Both the abacus and the "bones" provide excellent enrichment activities to begin an understanding of how early calculators were constructed, such as Pascal's "La Pascaline". The abacus also provides an introduction for an investigation into the binary system or base two and why base two was necessary in the development of the modern computer.

Another concept that many children have difficulty mastering is that of binary numbers, until it is compared to a switch than can be either "off" or "on". Models containing switches are used and children guess the numbers that are being represented. These models lead to a beginning understanding of how the computers work. Although students find this activity interesting, they are more interested in actually using a computer and finding ways to manipulate it to their benefit. Therefore, it is important to provide hands-on activities as well as related activities.

The computer has proven to be an interesting tool in developing mastery of another difficult skill for may students, that of understanding word problems. The basic computational skills of adding, subtracting, multiplying, and dividing could be demonstrated on the computer through developing word problems using the programming language of BASIC. In a very structured lesson, I led the students though many word problems by taking the problems apart and rewriting them for the computer. Our programs were fed into the computer and run. The excitement of using the computer to solve problems caused more interest in the work and improved problem solving skills. This tool was new to the students and, therefore, made the learning more fun.

Perhaps the most interesting way to use the computer in the math class is by using LOGO. LOGO is a programming language that teaches geometry in an exciting and meaningful way. The meaning of angles and directions are clearly understood by the students when they visually see them on the monitor. LOGO also requires a great deal of problem solving by students as they work to create pictures on the screen. They quickly learn the computer does exactly what they tell it to do, so writing clear directions becomes an important skill to develop. Children soon learn how to correct their directions when the desired picture does not appear. Teaching graph plotting also has added dimension when the computer is used. Again, learning to write exact directions is essential which requires attention to detail and concentration. Children are excited when they are able to create graphics on the computer. They have so much fun they are unaware valuable problem solving skills are being developed.

There are many other ways the

computer can be used in the math class. Commercial software packages are an excellent tool for reinforcing math concepts. There are many good packages available for this purpose. In my experience, however, students enjoy this type of activity only for a limited time and prefer the programming activities which allow them to be creative and in control of the computer. By using the computer to do basic math functions they realize it is another tool such as the calculator or typewriter and are not afraid of it when encountered in their daily life.

SNIFFING OUT GOOD SOFTWARE PROGRAMS: USING THE NOSES PROGRAM

Dr. Terry Northup

Wayland Baptist University
Plainview, TX

Dr. Gene E. Rooze

Texas Tech University
Lubbock, TX

This session will discuss the need for and the problems of evaluating software. Most evaluation systems are subjective, and of limited value for this reason. Therefore, the majority of this session will focus on the description, demonstration, and practice in the use of an objective software evaluation program titled NOSES. This will include group practice using the form to evaluate a software program.

There is a need to evaluate all materials used in education. First of all, ineffective materials waste time because they do not produce effective learning. The teacher, then, must reteach the material or move on to the next unit without the students learning the material. Another possibility, of course, is that unevaluated materials might effectively teach incorrect information. A final possibility is that poor material can kill motivation. Thus, poor material inhibits learning. Given the public's concern for improved learning on the part of students, teachers cannot afford to waste time using ineffective materials. In addition, there is the matter of cost, another concern of a tax-burdened public. It is a waste of scarce resources to purchase materials that do not produce results.

These concerns especially effect the purchase of microcomputer software. Many manufacutrers have rushed into the market with software and made exagerated claims. Much of this material is quite expensive compared to books and learning center materials which would accomplish many of the same purposes. Many of these programs are boring and teach inaccurate information. In other words, they can contain the same faults as other types of educational materials.

In addition, the evaluation of software presents several specific problems. Six of these which must be considered will be explored and discussed.

1. As a new media form, most teachers lack training and experience in evaluating software.

2. There are few accepted guidelines for evaluating software. To some extent, beauty is in the eye of the beholder. In addition, there are several different types of software programs and it is difficult to apply the same set of guidelines to them all.

3. In selecting books or other materials one can often rely on the reputation of the author or the publishing company. Since microcomputer software is such a new field, there are no established reputations on which teachers can cling.

4. While advertisements for books and other educational materials are relatively straightforward, ads for computer programs tend to be extravagant.

5. It is quite common for textbook companies to supply sample copies of their materials for evaluation. Software producers, wary of "piracy" seldom do this. Thus, it is rare that a teacher can see a program before buying it.

6. The entire problem is complicated by techinical questions which do not apply to other media. With few exceptions, all educational film is 16mm and will run on any school's projector. A cassette tape will play on any cassette player in workable shape. Of course, this is not true for computer programs.

One source of information about programs under consideration is published reviews of software programs. These are abundant. There are also several commercial sources of software evaluations, such as: THE DIGEST OF SOFTWARE REVIEWS and COURSEWARE REPORTCARD. In addition, Scholastic Magazine and others have created software evaluation forms for use by schools. Unfortunately, all of these evaluation systems are largely based on subjective judgments As such, the reviews report judgments of the reviewer, who is unknown to the teacher responsible for the purchasing decision. Can teachers rely on such information for important decisions? The presentation will give a brief look at these three systems and demonstrate their weaknesses.

There are two major problems with reviews and evaluations that are heavily based on subjectivity which make them an inadequate basis for decision-making.

1. Often the reviewer is anonymous. Even if the reviewer's name is given, a teacher will know little or nothing about the reviewer's background. For example, has he/she ever taught school? If so, has he/she taught near the grade level for which the program is designed? Is the person a computer genius or a typical consumer? (If a reviewer says a program is easy to operate, for whom is it easy?) How much faith can one put in the subjective comments of a stranger?

2.Even if one knows a reviewer is qualified, will his/her perceptions correlate with yours? Teachers seldom trust the subjective judgments of others on important issues. How many teachers would rather select the textbooks they must use rather than trust the judgment of the curriculum director? How many teachers trust the judgement of administrators when it comes to merit pay decisions? If teachers won't trust the subjective judgments of people they know, why should they trust the subjective judgments of reviewers whom they either don't know or only know by reputation.

Given a limited amount of money to spend on software, it is imperative that teachers spend these dollars wisely so that the programs they get are useable and meet their needs. In order to get useable information, information that allows the teacher to be the decision-maker, the information needs to be objective. Given the appropriate information, the teacher or committee making the software decision can determine whether a program in question will likely meet their needs. Then, if possible, they should review a copy of the program and test it out with their students.

In an attempt to create a system that would yield objective information, one of the presenters developed NOSES-Northup Objective Software Evaluation System. It provides a series of categories to which the reviewer responds with information about the program. The consumer can then determine whether the program meets his/her needs. You are encouraged to use the NOSES form, however it is requested that you identify it by name. The author would also appreciate a copy of the reviews you create.

NORTHUP OBJECTIVE SOFTWARE EVALUATION SYSTEM (NOSES)

IDENTIFICATION INFORMATION

Program Name:

Subject Matter Area:

Structure:

Computer System Requirements:

Networkable:

Price:

CONTENT

Grade Level of Subject Content:

Description--Goals or Topics Covered:

Reading Level:

Accuracy of Information Presented:

USEABILITY

Clear, Simple Directions:

Auto Engage or Manual:

Escape If Correct Response Is Not Known:

Amount of Typing Required:

DOCUMENTATION

Adequate Printed Instructions:

Support Materials:

Suggestions for Follow-up:

EDUCATIONAL TECHNIQUES

Reinforcement:
- Is there a variety?
- Is it appropriate?

Screen Report of Final Results:

Wrong Answer Response:
Hint?
Explanation of correct answer?
Branching?

MANAGEMENT-STUDENT RECORDS

Passwords:

Handling of Successive Scores:

Ease of Updating:

Retrieval:

Report of Time taken:

GRAPHICS

Variety:

In Content and/or Reinforcement:

Clear Message:

Action:

In the session, each of the sections of NOSES will be described and explained. Sample evaluations will be distributed and participants will have an opportunity to use NOSES to evaluate a piece of software in a group activity.

Learning How To Use Delta Drawing

Frank Piasecki, Ph.D.
Richardson Independent School District
Computer Instruction
1300 Cypress
Richardson, Texas 75080

The Delta Drawing software package by Spinnaker Corporation offers students in the elementary grades an opportunity to explore and become aware of computer capabilities through the world of art. Delta Drawing lets students draw full color pictures by using the computer yet requires only a minimum amount of typing. Most Delta Drawing commands are activated by pressing single keys on the computer keyboard. Delta Drawing is recommended for children ages 4-14 and features many significant computer literacy concepts.

Delta Drawing is a commercially prepared software package produced by Spinnaker Software Corporation. The use of Delta Drawing is suggested for Grades K-4 though it is likely that students in Grades 5-6 will find the use of this item interesting and worthwhile. Delta Drawing is being recommended for use because many significant computer and programming concepts are presented to students at an easier level appropriate for students in the lower elementary grades.

Delta Drawing allows students to explore various computer capabilities by drawing pictures on the screen. Many of the simpler level Logo concepts have been incorporated into Delta Drawing and the use of this item is encouraged for pre-Logo orientation.

Delta Drawing is especially easy to use as it has been designed in such a way where most command functions can be activated by typing individual keys. Only minimal keyboarding skills are required.

In order to work on Delta Drawing activities it is necessary for the disk accompanying this software package to be loaded into computer memory. Delta Drawing may be run on either the APPLE II+ or APPLE IIe computers. To run this item on the APPLE IIe, however, you must put the computer into the upper case mode. To do this just press the CAPS/LOCK key.

The complete Delta Drawing software package contains a helpful and easily understandable booklet called the User's Guide, a series of laminated step-by-step activity cards ideal for students to use containing 14 numerically arranged and varied Delta Drawing activities, and the necessary software disk.

Students can quickly begin to use Delta Drawing to develop pictures and/or patterns on the screen with minimal teacher instruction and direction. The laminated activity cards offer many of Delta Drawing's essential elements and the use of these cards by students should prove to be beneficial.

Like Logo, Delta Drawing has been developed to offer students a way of working with the computer by telling the computer how to draw pictures. The "Delta Cursor," a small triangular-like pointer, is the movable item which appears on the screen and leaves a visual trail of its movements. Students must instruct the Delta Cursor to move by inputting specific directions into the computer. The Delta Cursor will do only what it is told to do.

The use of Delta Drawing incorporates much more than just the sketching of a few simple pictures. Instructing the computer to use colors, storing instructions in memory for later use, saving Delta Drawing programs on file disks, printing out Delta Drawing pictures, and working with a wide range of Delta Drawing commands are just some of the possibilities available with this software package which offers students exposure to and familiarity with some very important and powerful computer concepts.

It is recommended that teachers first become familiar with materials included with the Delta Drawing software package before introducing Delta Drawing to students. The 14 laminated Fast Start activity cards are easy to use and not overly time consuming. Working through the activities on these cards should provide one with a good understanding of what is involved with Delta Drawing and what types of things can be done with it. The accompanying Delta Drawing User's Guide offers a comprehensive background into all of the special functions incorporated into this program. The use of this guide, as needed, is encouraged.

Getting Started with Delta Drawing

1. Load the Delta Drawing disk, follow screen presented instructions, and answer the introductory questions.

2. The Delta Cursor should appear in the middle of the screen. It is very easy to get the cursor to draw. Observe what happens when the following keys are pressed:

   D=DRAW (The cursor will draw a line in the direction in which it is aimed.)

   R=RIGHT TURN (The cursor will turn 30° to the right. Press R three times in a row and the cursor will turn 90° to the right.)

   L=LEFT TURN (The cursor will turn 30° to the left. Press L three times in a row and the cursor will turn 90° to the left.)

   E=ERASE (Pressing the E will erase the last command entered. Press E two times and the last two commands will be erased.)

   CTRL E= ERASE PICTURE (Press CTRL E and the entire screen will be cleared of any drawing and the Delta Cursor returned to the center of the screen.)

   M=MOVE (Press M and the cursor will move forward in the

direction in which it is aimed without leaving a draw line behind it.)

U=U TURN (Press U and the cursor will make a U-turn.)

3. Press CTRL E to clear off the screen. Practice using the above presented command keys. Make a square, triangle, house, etc. To make a "circle" press the D and R keys 12 times in a row until a closed figure appears on the screen.

Refer to the Fast Start cards and User's Guide to add color to your pictures and to take advantage of the computer's memory as you develop more complex procedures. Remember, many significant Logo concepts are incorporated into Delta Drawing. By using this software item younger students are able to take advantage of some of the computer's capabilities at an easy level with a minimal amount of typing.

TEACHING THE TEACHERS: THE KEY TO INSTRUCTIONAL WORD PROCESSING

Karen L. Piper
1108 Villa Downs
Plano, TX 75023
(214) 248-6300

In this article the author briefly describes a recent research project in which students used the word processing capabilities of the microcomputer to combine sentences and complete story starters. A major key to the effectiveness of the program seemed to be teacher training. The success of such an instructional word processing program may very well depend on teacher ability to: 1) recognize effective uses of instructional word processing, 2) learn how to use the program themselves, and 3) effectively introduce the program to students. The author provides tips for teacher training in these areas.

You have succeeded in convincing administrators and school board members that the word processing microcomputer can be used to enhance student writing ability. You have previewed and evaluated countless word processing programs and have secured the necessary hardware and software. In addition, you have devised a plan to introduce a variety of uses for instructional word processors. It seems as if your dreams have been realized. In actuality, your work has just begun! The success of your program may very well depend on your ability to teach the district's teachers 1) about word processing and its instructional uses, 2) how to actually use the word processing program you choose, and 3) how to introduce the word processing program to students.

When Bill Adkins, computer coordinator for Plano I.S.D., approached me concerning the possibility of conducting my doctoral research in the Plano district, I was ecstatic. Using Bank Street Writer* and Apple** microcomputers, I planned to deliver sentence combining exercises to sixth grade language arts students. Students would 'combine' the sentences, making use of the power of the word processing program and would expand their 'combinations' into creative writing projects.

Having designed a similar project in the spring of 1983 for Abernathy Elementary School (Abernathy, TX), I felt confident about the technique and materials I would use. However, Plano I.S.D. was allowing me to provide instruction to one-hundred sixth grade language arts students as part of a district-wide thrust to use the word processing powers of the microcomputer. Because the study was no longer small nor pilot in nature, and because teachers would eventually train sixth graders to use this tool, training the teachers became a foremost concern.

Outlining the Study

In order to help you understand the task at hand, some background information about the study itself might be useful. The pretest-postest analysis of covariance study runs for ten weeks, beginning in mid-January, 1984. One-hundred sixth grade students representing four heterogeneously-grouped, intact language arts classes comprise the population. Each of the four classes of twenty-five students represents one of the following research groups:

*Trademark of Apple Computer, Inc.
**Trademark of Broderbund

1) Sentence combining and story expansion activities delivered by word processing programs on microcomputers;

2) Sentence combining activities (no story expansion) delivered by word processing programs on microcomputers;

3) Story expansion activities (story starters and creative writing, no sentence combining) delivered by word processing programs on microcomputers;

4) Traditional sentence combining and writing expansion; microcomputer used for CAI and parts of speech instruction.

The purpose of this research is to: investigate whether independent or interactive activities in sentence combining and story expansion delivered by word processing programs on microcomputers can significantly effect growth in writing ability, reading comprehension, and attitude toward writing.

Following pretests, appropriate students receive instruction in sentence combining and/or story expansion exercises delivered on microcomputer. Treatments occur twice weekly for approximately forty minutes. Students spend 25 minutes working independently on the appropriate learning package and 15 minutes in discussion of their written products with the teacher or edit team mate.

The first session each week for the group representing 'total effects' focuses on teacher introduction of sentence combining technique and written production. The second weekly session focuses on editing and story expansion. The other two experimental groups work on text files appropriate to their group affiliation. For instance, the students in the sentence combining/microcomputer group complete the sentence combining activity on the computer and edit or revise the resultant writeouts. The story expansion/microcomputer group receives a story starter made of sentence combining writeouts similar to the clusters produced by the total effects group. These students read and expand the set of sentences into a story, then edit and revise the story. Writing products are saved on diskette (and ultimately printed out) and are edited by student edit teams.

The ten-week timeline allows for two weeks of keyboarding practice. Teachers then instruct students in the use of the Bank Street Writer through guided practice sessions. Eight weeks of actual treatment is necessary to realize any growth in writing ability or reading comprehension as evidenced by previous studies (Piper, 1983; 1984).

Teacher Role

As evidenced by the summary of the study, the role played by the teacher is critical. The teacher must present the instructional material, schedule time wisely, encourage edit teams and student writers, and serve as a resource person for the use of the Bank Street Writer.

To help minimize teacher difference and provide adequate background, each teacher receives learning packets with standard materials, scripts for lessons (including what to say, how to guide discussion, presentation materials), the actual lessons, and tips for motivating and guiding student revision and expansion activities. Teachers deliver instruction using learning packets and text files I designed for use with Bank Street Writer.

Teaching the Teachers

Most of the teachers involved seemed to have little problem with the actual instructional material. Each lesson and accompanying script was presented to them in 'hard copy' for their use. A lesson consists of learning objectives, introductory scripts, scripts for presentation of the material, and the actual lesson the students view on the computer. The anxiety most teachers experienced concerned the use of the word processing program. They knew that

they might very well have to 'troubleshoot' for students who became confused or unsure of their ability to use the program. In order to allay some of these anxieties, I worked with the Plano I.S.D. to design and present instruction in Bank Street Writer.

Key sixth grade language arts team leaders attended a full day of inservice, during which time I delivered background information concerning the instructional use of word processing, the use of sentence combining and writing expansion with the word processor, and actual instruction with the Bank Street Writer.

To provide this instruction, I designed two learning packets. The first packet detailed instructional word processing, background information, and research bases. Information concerning sentence combining theory, use, and research was also included. In addition, several pages represented actual sentence combining lessons that the teacher would use to present materials to students.

The second packet was a learning module I designed for instruction in word processing with the Bank Street Writer. This packet included brief background, learning objectives, simplified instructions concerning the various modes and keystrokes, and sample 'hard copy' exercises teachers used to practice typing and editing abilities.

Teachers went to the computer following a brief introduction. They worked through the module in a 'guided discovery' manner. In addition to the packet, I had prepared a data diskette for each teacher. We loaded the diskette into the computer for 'on-line' practice. It contained three separate text files to allow for various kinds of practice with Bank Street Writer. The first two exercises consisted of entering brief text, making specific corrections, saving and loading, and toggling between modes. The final exercise served two purposes. This activity was an open-ended evaluation of the instruction the teachers had received. Teachers typed in their ideas to complete the stems I had supplied. To do so, they had to be able to load the diskette, choose the proper mode, position the cursor correctly, type in text, and make appropriate edits before saving and printing the evaluation.

After approximately two and one-half hours, teachers turned in their data diskettes for printing. The evaluation forms were printed out to help design other instruction or practice and to pinpoint potential problem areas. Each stem on the evaluation form and its most common responses appear below. Teacher response to these items might prove useful to you in designing similar instruction for teachers in your district.

1. The thing I like most about the idea of word processing is...
   - "having printed material..."
   - "it is new and exciting"
   - "it is much easier to correct mistakes"
   - "getting the students to write"
   - "increased editing ability"

2. I am a little concerned about...
   - "working with low level students"
   - "being able to teach others"
   - "teaching it to students"
   - "erasing everything"

3. I could use some more information about...
   - "teaching this material to others"
   - "editing..."
   - "the time involved in becoming proficient"

4. The most interesting thing I learned today was...
   - "that I am already catching on!"
   - "how to really use BSW"
   - "how to use all the modes correctly"
   - "real hands-on practice"
   - "learning how to work this machine!"

5. Information I needed but didn't recieve...

"how to organize my classroom physically"
"how to schedule students on the computer"
"I just need more practice!"
"what we need is more time to practice on our own"

It can be noted from responses that most teachers were convinced of the instructional value of word processing and excited about its possibilities. They were beginning to believe that they could use the computer as a writing tool, but were still concerned about their ability to teach the word processing program to other teachers and students. Opportunity to practice using the program was an often-expressed concern.

To alleviate these concerns, several steps were taken. Opportunity to practice was planned; teachers were allowed to check out both a computer and word processing program over the winter break. One of the other major concerns, scheduling and physical set-up for individual classrooms has been addressed as well. Because the actual logistics of physical arrangement and scheduling are situation-dependent, visits to individual schools are planned to help assist teachers and alleviate these concerns.

Following the instructional session and a review of the responses teachers had made on the final exercise, I spent some time outlining steps they might take in introduction of this word processing program to the students. These steps include:

1. Learn the program well. Spend time actually using it.
2. Try using the program without the manual to pinpoint possible trouble spots.
3. Make a list of the major functions and try to implement each one without the documentation.
4. Make a 'quick key chart' as a student reference tool.
5. Choose and train key students as aides before introducing the program to the entire class.
6. Train students using: tutorials on disk, guided activities for student manipulation of text, and text files written with specific learning objectives in mind.
7. Present the program to the entire group, making use of trained student aides.
8. Allow time for constructive practice.

Conclusions

Word processing programs can be a a valuable tool for writing instruction. However, the effectiveness of your program may very well depend on the teacher's ability to use the word processing program and to teach students how to use the program. By planning for appropriate instruction in these areas, both teachers and students will benefit.

REFERENCES

Piper, Karen. Word processing in the classroom: using sentence combining exercises with elementary students. Conference proceedings of the National Educational Computing Conference. Baltimore, Md. June, 1983.

Piper, Karen. The electronic writing machine: using word processors with students. The Computing Teacher, vol. 11, no. 5, Dec./Jan., 83-84, pp. 82-83.

HOW TO CREATE 35mm COLOR SLIDES WITH GRAPHIC SOFTWARE AND YOUR HOME COMPUTER

Gerald Pollard
Texas School for the Deaf
Box 3538
Austin, Texas 78764

This presentation explains how computer graphics software can be used to create the graphic of your choice (limited mostly by your artistic ability) and how 35mm color slides can be made from your video screen.

COLOR GRAPHICS

There are a variety of ways to create color graphics with your home computer. This brief discussion of graphic software will be organized by direct keyboard entry and external plug in boards with an understanding that some graphic packages utilize both input modes.

KEYBOARD. One easy way to create color graphics, especially text, is to utilize the graphic commands like GR, Color = __, HLIN, VLIN, and PLOT. These commands are listed in the Apple Basic Programming Manual. They are easily taught to students and provide a stimulating and almost immediate way of creating colorful graphics. Using LOGO programming language is another way to produce graphics, especially geometric shapes. Logo can be easily learned by students and adults.

EXTERNAL MODE. There are several different brands of graphic tablets that connect to the Apple via a slot card. These tablets allow one to draw freehand, trace, or create circles, squares, boxes, lines, etc. from a board menu. These boards are usually used by serious users, artists, draftsmen and are somewhat expensive ($350 to $750).

A recent product is the Koala Pad. It consists of an external pad and connects to the game port of a computer. It is simple to operate and provides an easy way to create colorful graphics. Preschoolers as well as post-college age students enjoy using it.

The above software provides ways to create graphics but does not allow text to be placed on top of the graphics, which is a frequent need in making title slides. Use Flying Colors, by the Computer Color Works, the Graphic Magician by Penguine Software or Apple Mechanic by Beagle Brothers to accomplish this feat. For a more comprehensive review of graphic software, see "Apple Graphic Software" in the February issue of Creative Computing.

PHOTOGRAPHY. Our challenge is to create the desired color graphics and then make a 35mm color slide from the T.V. screen. This is not a difficult project but it does require the necessary equipment and adherence to certain rules. I've found the following to be helpful.

EQUIPMENT AND FILM NEEDS. You will need a 35mm single lens reflex camera with a normal or telephoto lens, a built in light meter, a tripod, cable release and color correction filter, and high speed daylight color film (ASA 400). Since this film is extra sensitive to blue-green, use a color correction filter (CC 40R [red]) to reduce these hues.

PROCEDURE. Run the graphic program saved to your disk and then adjust your set as to color balance, contrast, focus, etc.

Postition your tripod and camera in front of your T.V. so that you view only the T.V. screen and not the T.V. housing as you look through the lens.

Set your shutter speed to 1/8 of a second. A slow shutter speed is needed because the electronic light beams of T.V. image travel at 1/30 of a second. Shooting slower than 1/30 of a second insures that all scans will be recorded on the film.

Due to the reflective nature of the T.V. screen it is wise to turn out all room lights and put the back of the T.V. closest to any window or light source. One should not see any reflective glare when looking through the camera lens. Glare can be eliminated by positioning a black cloth or cardboard to block the light source.

Turn your light meter on and adjust your meter accordingly. To make the perfect shot, bracket your exposures. That is, take a picture with your meter in the exact middle, then one F stop above and below that setting. This is necessary because T.V. light sources can confuse some meters. When all is ready, use your cable release to expose your perfect picture.

Creating title or instructional slides via computer graphics can be a creative task and an economical way of producing useful color slides.

THE HISTORY OF THE COMPUTER: A MODEL FOR INFUSED OBJECTIVES IN SOCIAL STUDIES

Rita Potter,
Corpus Christi Independent
School District

The history of the computer like all developments in the complex history of mathematics, science and philosophy, is most easily comprehensible to the student when placed in a established frame of reference. The major curriculum decision which should made before teaching the history of the computer must include consideration of the essential elements in other academic subjects. The program alternatives include teaching computer literacy objectives according to the three models: units of study, infused objective and special courses. Economics would seem to be the appropriate strand to use when infusing the history of the computer into the social studies curriculum.

"It is unworthy of excellent men to lose hours like slaves in the labors of calculation." Gottfried Wilhelm Von Leibniz (1646-1716)

The growing emphasis on the development of computer literacy has presented educators with many complex problems involving content, placement, and implementation of an appropriate curriculum. The models proposed in the H.B. 246 discussion documents include the use of units of study, imbedded or infused objectives, or separate courses. The history and develpment of computers involves subject matter which is closely related to the proposed curriculum content of the social studies curriculum. The study of economics, an essential element in the course content of all required social studies courses may be used as an unifying strand. The study of inventors, inventions and calculating devices in the elementary grades could be extended into the secondary curriculum with a greater depth of treatment, including the philosophy and mathematical implications.

The history of the computer becomes as esoteric study when considered only in the context of technological development. No matter how absorbing to the initiate, the study of the generations of computers, with all of the sophisticated mathematical concepts, computerese jargon, structure of the electronic superbrains and design marvels presents a challenge to the teacher and student in a 'typical' class.

Throughout history, human beings have been fascinated with the magic of numbers. On a practical level they have manipulated and calculated numbers. The prehistoric hunters began with simple finger and hand signals, many of which are still prevalent in aboriginal cultures. The widespread use of a base ten number system may be attributed to this early

development. The first tallying device, a handy pile of pebbles, is acknowledged in the term, calculus, derived from a Latin word for stone. Even the earliest developments of religion and philosophy show this continuing intellectual fascination with the power, logic and order of mathematics. Within the social studies curriculum educators may use this aspect of human history to show the relevance of the development of the computer to man's past, present, and future achievements.

The Age of the Enlightenment and the development of the scientific method of inductive reasoning established the intellectual foundations of a new age in which science and technology could join in the partnership which later resulted in the Industrial Revolution. During the nineteenth century Charles Babbage emerged as the first man to conceptualize a true computer although he was unable to complete his devices because the craftsman of his day were unable to develop the necessary precision tooling. Babbage attempted to combine the practical application of the Industrial Revolution with the love of mathematical reasoning of the Enlightenment. His "failure" was more important to later developers than the facile success of other inventors.

There are innumerable examples in the history of computers which may be used to illustrate the importance of intellectual achievement as opposed to mere mechanics. In developing this awareness the social studies teacher is preparing the student to understand other complex issues in the age of modern technology. By infusing the teaching of computer literacy objectives into the general social studies curriculum, educators may develop an increased awareness of the importance of logical reasoning and mathemical understanding in all phases of human achievement, particularily in economic progress.

BIBLIOGRAPHY

Bell, E.T. Men of Mathematics, New York: Simon and Schuster, 1937.

Bergamini, David, et. al. Mathematics. New York: Time-Life Books, 1970.

Cupertino Union School District. "K-8 Computer Literacy Curriculum". The Computing Teacher, Vol. 10, No. 7, (March, 1983), 7-10.

Golden, Frederic. "Big Dimwits and Little Geniuses," Time, Vol. 121, No. 1, (January 3, 1983) 30-32.

Horn, Carin E., and Poirot, James L. Computer Literacy: Problem-Solving with Computers. Austin: Sterling Swift Publishing Company, 1981.

Richman, Ellen. Spotlight on Computer Literacy. New York: Random House, Inc., 1982.

Rusch, Richard B. Computers:

Their History and How They Work. New York: Simon and Schuster, 1969.

Walker, Roger S. et. al. Understanding Computer Science. Dallas: Texas Instruments, Inc., 1981.

Student Use of Data Management Program: New Tool for Extended Study Skills

Dannah Procter

Box 1070
Freer, Texas 78357

A data base management system can be used by secondary school students to enhance and extend study skills such as note taking, organizing information, catagorization, etc. By designing a file and entering and retrieving information from it, students can order, classify and arrange the new concepts they acquire in a subject area. Going beyond merely storing information, students must examine the relationship between concepts and facts in order to search and retrieve their data. In the process, they become computing literate.

One of the most commonly heard topics of debate in educational computing circles is that of computer vs. computing literacy. The former position is often equated with knowing about computers and programming while the latter is associated with stressing the use of computers for different applications. On the one hand, a student who is considered computer literate can do some simple programming, can load and use software, and is knowledgeable about computer history, careers, and the impact of computer use on society. The other stance is that the student who can use a computer to accomplish a desired task is literate: One uses computers as tools for problem solving, ergo, one is a computer literate person.

If the computing or using position is accepted, how is it put into practice? What opportunities exist? Word processing is obvious, but what other major uses of computers are common? Two come to mind, electronic spread sheets such as Visicalc[tm] and information management. Spread sheets can be used for various applications within the secondary students's curriculum, and should be taught to good advantage. What about information management? What kinds of data do secondary students have to manage? The answer is overwhelmingly obvious: They have to organize and assimilate all the information they acquire as part of their classes. History, chemistry, literature, government, economics, biology, etc., all have heavy loads of concepts and facts to be comprehended. Students have to see beyond the masses of facto the underlying academic structure. Somehow--with the assistance of skilled teaching--students must visualize and

understand that structure. A data management program will assist students in that understanding.

In order to use a data management system, certain conditions must be in effect. First, students must have access to computers, must have adequate time allotments to use them for study purposes, and must know how to use both a computer and a data management system. It is conceivable that all these requirements could be met through a computing literacy course that provided students with access to machines, training to use them, and the time and motivation to use them for practical purposes. It is also feasible that the subject matter teacher who has occasional access to computers could teach the students the skills and provide the time necessary to create the student data bases.

A few example applications of data management programs in specific subject areas should serve to show the potential scope of uses. In literature, for example, the student or small group of students working together could design a data record that included the following fields:

LITERARY WORK
AUTHOR
DATE
THEME
PROTAGONISTS
CLIMAX/RESOLUTION

One desirable result of using the data management program is that it forces the student to be consistent and precise in naming and entering data. The user who tries to print all the entries under the theme of "revenge" quickly learns that any that have been labeled "vengeance" will not be included. This is an important lesson for sophisticated data base use.

A history record might consist of a biographical file that included the pertinent name, dates, political party and significant contributions. Or it might consist of entries on a multiple-strand time line such as date, item, and strand, with social, industrial, political, cultural, and military being possible strands.

Biology has many potential records, the simplest being one that classifies entries from kingdom down to species; or records might consist of the species along with its habitat, food sources, enemies, etc.

Even based on these brief examples, it should not be difficult to think of other applications in these and other subject areas. The question becomes not how to do it, but why to do it. Why have the student use the computer to do something that can be done with paper and pencil in a notebook?

Obviously, the study value comes, not from merely recording the data,

but from analyzing the subject matter in order to extract the information. Equally obviously, the data could be stored on written forms calling for the same information. However, the motivational impact of using the computer for the data storage is a powerful incentive to do the requisite study necessary to acquire the data. It is simply more fun to store the information in the computer than it is to write it by hand. A second motivating factor is the visual quality of the information that can be recalled from the system. Instead of messy, illegible notes, a printed professional quality report is generated.

Two other considerations should be mentioned. First, the computing literacy position requires the student to use the computer as a tool for solving some problems. This use as a study tool provides such a purpose and can contribute to the correlation of different subject areas by coordinating the work done in the computer literacy class with other subject matter areas. Secondly, one of the fastest growing areas related to personal computing is communications, using the computer along with telephone lines to communicate with other individuals and agencies outside the home. It is predicted that on-line data bases will be an increasingly important information source in time, perhaps, even replacing the newspaper. It is necessary to have some skills in data base searching in order to use these programs. Learning how a data management program works--even a small one such as is suggested here--is a preliminary step toward greater communication literacy. Yesterday we taught students how to use the card catalog, today we teach them how to use data bases, tomorrow we teach them to use on-line information sources. With the information explosion we are experiencing today, no one can be expected to begin to stay abreast of all the new data even in a limited field. Our new goal must be to be able to locate the information we need as we need it. Students must know how information is classified and stored in order to function successfully in today's world. This will me even more true tomorrow.

By having students use data management programs, we are teaching them to structure the information they are acquiring in their classes, to use information storage systems, and giving them prerequisite skills for using the on-line information bases they will need in the future.

Planning For Computer-Fluent Teachers

Dr. Leah Rampy
Dr. Barbara Sadowski
Dr. Barbara Signer
Department of Curriculum and Instruction
College of Education
University of Houston, UP
Houston, Texas 77004

The Texas Education Agency (TEA) has proposed that a three-hour course in computer literacy be required for teacher certification. This proposal raises questions about the type of computer instruction most appropriate for pre- and inservice teachers. This paper includes a discussion of several issues raised by the TEA proposal and describes a graduate program aimed at preparing computer-fluent teachers for tomorrow's classrooms.

The need for teachers to become computer-fluent is indisputable. The move to require computer literacy education for students in grades K through 12 is accompanied by a demand for teachers who are able not only to use computers for instructional purposes, but who are able to teach their students how to use computers. Teaching students to use computers requires computer-fluent teachers.

Computer fluency is the last of three stages of learning about computers. In the first stage, computer awareness, the individual becomes knowledgeable about computer applications and their effects on society. In this stage, the learner may begin to see ways in which computers can be used in education. Computer literacy, the second stage, involves personal use of the computer as a tool. The individual is able, for example, to use the computer for word processing or to run software packages. The computer-literate individual will have a general knowledge of programming in at least one language. The third stage is computer fluency. Fluency implies a degree of effortlessness and polish beyond that of literacy. Although we may consider a person literate who reads and writes at a fifth grade level, we may not consider that person fluent in reading and writing. Similarly, a computer literate person does not have the additional skills of someone who is computer-fluent.

Planning must begin immediately if we hope to encourage computer fluent pre- and inservice teachers. This paper will explore ways in which we can prepare computer-fluent teachers. The first part will include suggestions for offering computer instruction to preservice teachers; the second section will outline one means of increasing computer fluency for inservice teachers.

Preparing Preservice Teachers

A proposal by the Texas Education Agency (TEA) is presently before the State Board of Education to include a three-semester hour general education course in computer literacy as a part of teacher certification requirements. If approved, this requirement would mean that preservice teachers would receive instruction in "computer-

related terminology and applications, the history and development of computer in society, using the computer as a tool, communicating instructions to the computer, future trends, and computer-related careers." These requirements are virtually identical to those proposed for students in grades K through 12 by TEA. The proposed state requirements for teacher certification and for K-12 curriculum change raise several issues about the preparation of teachers now and in the future.

Who Should Provide Computer Instruction for Preservice Teachers?

If a three-hour "computer literacy" course becomes a part of the general education requirements for preservice teachers, who will provide the instruction? This issue was discussed at a conference sponsored by several special interest groups of the Association for Computing Machinery in June 1982. Educators at this conference suggested that an existing computer science course be modified to provide a general introduction to computer uses or that a teacher-oriented computer course be offered by either the computer science department or the College of Education. We submit that such courses should be the responsibility of Colleges of Education. Courses offered by the College of Education, however, may not be accepted by the state as meeting the requirement for a general education course.

Although an argument can be made for using the expertise of faculty members in computer science departments, there are several problems with delegating the responsibility. Computer science courses are already over-crowded on most campuses. Computer science faculty are understandably more interested in meeting the needs of their own students and, historically at least, have not been eager to modify courses to fit the needs of educators. More importantly, however, preservice teachers have unique needs. For example, the introductory computer course in the College of Education at the University of Houston includes work with Logo and Bank Street Writer--knowledge important to future teachers which typically would not be discussed by computer science departments. Additionally, we work with preservice teachers on evaluating educational software, a need specific to educators. Although computer science departments may offer general courses appropriate for preservice teachers, instruction in the use of computers as a tool for teaching and learning is still necessary and the expertise for that instruction lies within the College of Education.

How Should Computer Instruction Be Provided?

Computer instruction for preservice teachers could be presented as a course, a series of courses, or integrated into required methods courses. If the proposed state requirements become effective, a minimum of a single course would be required. A single course is an effective means of delivering general instruction; however, a single course is insufficient to prepare computer-fluent teachers. Integrating computer instruction into existing methods courses would be beneficial in that it would use the expertise of subject matter specialists, especially in the evaluation of software. Presenting computer instruction within various methods courses assumes that faculty who teach these courses are themselves computer fluent, a tenuous assumption at best.

Given the likelihood of a required general education computer literacy class, the need for College of Education faculty to be involved in the education of teachers, and the importance of utilizing subject-

matter specialists, a three-pronged approach may be best. First, a general education course outside of the College of Education would be offered to preservice teachers; the goal of this course would be to create teachers who are computer-aware. Second, a course offered through the College of Education would focus on the educational applications of computers. Finally, methods course instructors who are computer fluent would be responsibility for teaching students about the use of computers in a given subject area. This approach should insure that all students have a general knowledge of computer uses as well as an understanding of how the computer can (and cannot) be used in specific subject areas.

What Should Be Taught?

The proposed state-requirement referred to earlier suggests that the general education course should provide what Moursund (1983) calls a "talking-level of knowledge." Moursund argues that computer literacy involves "doing." In the future what will be important is "knowing how to utilize the capabilities of a computer and integrating this knowledge into one's overall knowledge and performance" (p. 3). Both talking and doing are important for the preservice teacher and computer instruction should reflect both. The curriculum developed for preservice teachers now, however, will be a short-term one. The type of course proposed for certification will not be necessary when preservice teachers have received computer literacy instruction at the pre-college level. However, these students will not reach education courses until about 1988, so present computer literacy definitions for teachers will suffice for the short term.

In summary, the TEA proposes a minimum of a three-hour course in computer literacy for certification. Colleges of Education may be best suited to offer such a course because of the unique needs of preservice teachers. Although a single course may provide a solid framework for computer literacy, it will not be sufficient to prepare computer-fluent teachers. Finally, the curriculum developed now must eventually change to account for preservice teachers who receive computer instruction under state required computer literacy curricula in K-12.

Preparing Inservice Teachers

The proposed state requirements for computer literacy in grades K through 12 point to the need for teachers who are computer-fluent. The proposed curriculum for grades 7-9, for example includes developing flowcharts, writing "reasonably structured" programs, evaluating software, and studying the history and development of computers. Most school districts have developed inservice programs which have increased the computer awareness of teachers and administrators. However, due to limited time, staff, and in some cases hardware, inservice education alone cannot meet the challenge of preparing computer-fluent teachers. In order to aide in preparing computer-fluent teachers, the College of Education at the University of Houston, UP, offers masters and doctoral level programs in instructional technology. The following provides a summary of the masters level program.

Goals and Requirements

The master of education with a major in instructional technology is designed to serve the needs of two

groups: 1) teachers who wish to integrate the use of technology with instruction, and 2) students without K-12 certification who expect to use instructional technology in business training or instructional positions outside the K-12 classroom.

Certified teachers enrolling in the instructional technology program are encouraged to focus on developing specific skills in using technology as a tool to enhance instruction. Elective courses supporting the technology emphasis should focus on either curriculum or a specific curricular area. Courses outside of the college may be taken either in the curricular area or in a related area of technology such as computer languages, video production or management information courses.

Non-certified teachers choose electives in education which complement their background and provide a foundation in understanding the teaching/learning process. Curriculum courses in generic teaching strategies, developing curriculum, evaluating and improving instructional programs are strongly recommended. All students with an Instructional Technology Major are required to take 9 hours in research, learning theory, and social foundations and 9 hours of Applied Instructional Technology, Instructional Design, and Instructional Applications of the Microcomputer.

Description of Selected Courses

The following courses are offered to provide masters students with a degree of expertise in microcomputer applications to education:

CUIN 6320: Computers in the Classroom An introductory level course for elementary and secondary teachers focusing on developing a computer literacy curriculum, evaluating software, using teacher-authoring languages, and programming in Logo, PILOT, and SUPERPILOT.*

CUIN 6345: Instructional Applications of Microcomputers - A beginning course for educators in BASIC programming on the microcomputer focusing on algorithmic thinking for solving problems. Students in this course develop software for learners in elementary/secondary school.

CUIN 7345: Advanced Instructional Applications of Microcomputers - Solving problems in computer applications, using hi-resolution graphics, music, sound, text files and the development of simulations for instruction.

CUIN 7307: Structured Programming - A course using list process LOGO, KARL and Robot, the Pascal emphasizing structured programming techniques.

CUIN 7348: Seminar in Microcomputer Research - A seminar examining the literature on the effects of instruction using microcomputers and conducting a research study using teacher developed software.

In addition to those listed, a wide variety of courses are available to meet individual interests and needs. An interactive video course provides students with experience in developing interactive video for instructional purposes. Courses are offered in beginning and advanced computer graphics, administrative applications of microcomputers, and developing and marketing software for teaching in the language arts area.

If the proposed TEA requirements for the K-12 curriculum are to be implemented, computer-fluent teachers will be required. Inservice education, important for developing computer awareness, is not sufficient to prepare teachers who must assume responsibility for computer instruction. The Instructional Technology major at the University of Houston is designed to encourage the development of

computer-fluent teachers for tomorrow's classrooms.

REFERENCES

Moursund, D. (1983). Computer literacy: Talking and doing. The Computing Teacher, 10(8), 3-4.

Rogers, J. B., Ed. (1983). Computer education for colleges of education. New York: Association for Computing Machinery.

Social Studies, the Computer and H.B. 246

Gene E. Rooze, Professor, Texas Tech University
Terry Northup, Professor, Wayland Baptist University

ABSTRACT

This paper examines the use of the computer as a mediating device useful in the implementation of H.B. 246. It shows how the dominant views of social studies can be served through the use of drill and practice, games, tutorials, databases, and simulations. This article concludes that knowledge of results, interactive effects, motivation, application of skills and simulations available in present computer software can meet many of the goals of the social studies curriculum.

The microcomputer is a powerful mediating device which can be used to implement the social studies component of H.B. 246. This paper discusses: first, the effects the choice of a mediating device has on instruction; secondly, the views of social studies proposed in the new curriculum; and third, the kinds of software presently available to fill these goals.

In order for you to follow the discussion I'd like to review the "instructional interaction," model. You may recall that an accepted definition of curriculum is: those planned experiences obtained under the auspices of the school. Those experiences involve subject matter: knowledge, skills and values. This is the social studies content we are attempting to transmit to the student through a set of experiences we call instruction.

We speak of instruction as though it were a single element, but in fact it consists of: a set of teacher actions (TA), some mediating device (M), allowing a planned set of student actions (SA), which produce the resultant outcome.

Content → TA + M + SA → Resultant Outcome

A significant point is that changing any one of these components influences the outcome of learning. Therefore, the selection of a mediating device does make a difference. It effects both the teacher actions and student actions. Each mediating device: programmed learning, textbooks, films, or microcomputers affects the resultant outcome.

Possibly the best illustration of this principle is the use of the textbook as a mediating device. There are several advantages to the textbook. First, textbooks are cheap because they can be used repeatedly. A second advantage is that they contain information and commonly, student activities that can be used by teachers of varying skill levels. A beginning teacher can use the teachers guide, verbatim, and get similar results as compared to an experienced teacher. Third, textbooks allow a standardization of presentation that can be built upon from year to year. And finally, they are readily available since they are the traditional medium of instruction.

There are several curriculum concerns that make the textbook a poor me-

diating device. First, it calls for an active teacher role and a passive student role. Second, the use of a textbook often involves the memorization of facts, concepts, and principles. The way it is frequently used allows little analysis or creative thought. Third, the textbook approach can transmit the idea that knowledge is fixed--the exploration of new knowledge and the solution of social problems should be left to the experts. We have long accepted these weaknesses because the skill goals of social studies have never been a major emphasis. What about the use of the computer in social studies?

Since its birth early in this century, the overriding goal of the social studies has been the training of good citizens. But the views of how to train good citizens have varied. Presently, two dominant positions exist: first, that knowledge alone produces a good citizen and second, that teaching the citizen to think will do the trick. Proponents of the knowledge approach to social studies believe the best way to train a citizen is to provide him knowledge about his cultural heritage and teach him the values of the dominant culture. Advocates of the knowledge approach represent opposing ends of the educational spectrum. There is one group, the traditionalists, who promote the great literature, history and ideas of western man as the best curriculum. On the other end of the continuum are those who represent the academic disciplines of the social sciences. They believe the proper curriculum is selected from portions of the social sciences. The common assumption of both is that knowledge alone has value, and this knowledge produces a good citizen.

The second popular approach is called the skills approach. Advocates of this approach believe that the best way to train good citizens is to teach them a standardized set of skills. This set is intended to allow students to gather and organize information from which they can draw their own conclusions. The emphasis of this view is on the teaching of the processes of decision making. Advocates of this approach to social studies have recently also divided into two groups. The academic skills camp, described above, and the processes of the social sciences group. The latter believe that we need to teach the scientific approach. The essential argument separating these two groups is simply the set of skills which need to be taught.

Both groups maintain the view that present knowledge and values are useless in a changing, technological world. We need only teach future citizens ways to process information which will allow them to select from viable alternatives.

The changes proposed in H.B. 246 reflect these dominant views and attempt to integrate the two philosophies. But a careful analysis of the revisions inserted in the October draft of the "Social Studies Descriptors" indicate that thinking skills are becoming more important. The remaining portion of this paper will discuss the types of computer software which can play a significant role in implementing the proposed knowledge, skills, and values.

Drill and practice. The use of the computer as a drill and practice is common. The computer works in much the same way as a workbook except that it is interactive and provides knowledge of results very quickly. Cohen (1982) found a plethora of "geography drill" programs. Such software permits students to test their knowledge against that contained in a computer program. It provides a supplement to classroom instruction. Cohen found no programs which allowed practicing other social studies skills. Some use of reading software may be helpful.

Alesandrini (1983) reports that one of the problems with drill and practice programs is the way they are

used. Drill and practice programs are intended to be used after one has been given the initial instruction. Children are often assigned this software before such instruction is provided. The result is confusion and frustration.

Games. Some educational programs are designed as games. This technique which requires a player or players, a set of rules, a contest, and a winner is an effective means of teaching facts or mastering skills or concepts. Clement (1981) says educators would do well to gain an understanding of "what it is that will keep people entranced by computerized games for hours at a time. Very often they are developing skills of planning and strategy that instructional designers would find difficult to emulate by traditional methods."

Tutorials. Tutorial lessons can be used to supplement or even replace the direct teaching of a concept of skill. This could be an extremely useful tool to provide basic instruction in those situations where children need specific concepts or skills to complete certain tasks. In her most recent review, Cohen (1983b) identifies 3-4 which are concept oriented. Again nothing exists in the social studies skills area to date.

Databases. Several telecommunication services which provide a broad range of information (EPIE, 1983) are available for use with the microcomputer. They provide encyclopedia, almanac, and bibliographic services, similar to libraries. They are fast and relatively inexpensive. Databases are excellent supplements to textbooks because they contain up-to-date material. The use of student prepared databases are possible too.

Basing her ideas on the teaching of academic skills, Hunter (1984) suggests that children develop their own databases. She illustrates the use of pfs: File published by Software Publishing Corporation designed for Apple computers (similar programs are available for other systems). Data bases can be developed on states, countries, and even class members. Deciding what data to gather, gathering it, organizing it effectively, and drawing conclusions from the stored data would provide excellent skills teaching, practice and application.

Simulations. Simulations are based on a set of concepts and generalizations called a construct or model designed to describe real world phenomena. They can be used to provide students an understanding of a situation or event; or they can be used to teach the application of skills.

The potential of simulation has yet to be realized in the public schools. In the past, it took a lot of time to run a simulation because there was usually data to organize, and often statistically evaluate, before the next session could begin. The computer's speed has eliminated this problem. It has greatly simplified and diversified the use of this instructional method.

One misuse of the method must be corrected (Cohen, 1983a). If students are only allowed to operate within the construct and are not permitted to understand the elements of the construct, the simulation simply becomes another game.

Skills training using simulation is challenging. Imagine being able to test your decision making power against other individuals or against a computer program. The analytic use of simulation produces excellent results.

An assessment of the analytic qualities of existing simulations produces differing conclusions. Nilson (1983) feels that present simulations fail in at least two respects. They do not force players to develop and test strategies; nor do they provide experiences which will allow students to practice basic skills. An article

in the same journal more or less proves that "beauty is in the eye of the beholder." Friel (1983) uses the often criticized Lemonade simulation and does a masterful job of integrating several of the scientific skills: observation, data gathering, making inferences, forming hypothesis and experimentation.

The problem may be that many educators feel that the computer is a stand alone device: they should be used without the intervention of classroom instruction. It won't work with simulation. Early research in the use of the technique indicates that students must be prepared to run a simulation; they should then run the simulation; and then they need to discuss what they've seen. This latter step can be used to develop conclusions or to formulate testable hypotheses for a next round of play. Friel uses this method and obtains outstanding results.

The use of the computer can aid in the implementation of H.B. 246 in 5 ways. First it provides immediate knowledge of results. Immediate knowledge of results is present in both drill and practice and tutorial programs. This element is missing in most social studies classrooms, and children want to know "is that right teacher?" Second, the interactive effects created by this medium will likely make social studies more fun and even possibly more productive in teaching students the desired concepts and skills. Third, the use of educational games as drill devices is an accepted motivational method; most children enjoy contests they can win. Fourth, the use of databases, either commercial or student generated, will allow the application of skills, a long sought goal of social studies classroom instruction. And fifth is the important instructional technique or simulation.

Simulations are a natural part of social studies. They require the use of learned skills. Simulations also teach constructs, combinantions of concepts and generalizations, which explain the real world. No other single instructional device can run simulations as well as computers. This device will place the student in a more active role. It will also require the teaching of the skills of problem solving and decision making which are not accomplished in the present curriculum. If the social studies goals or H.B. 246 are really to teach both knowledge and skills, then the computer should be a useful tool.

References

Allessandrini, Katherine L. How Can Educators Spell Relief? C-O-M-P-U-T-E-R. Educational Computer Magazine, 1983, III (September, 1983), pp.32-35.

Clement, Frank J. Affective Considerations in Computer-Based Education. Educational Technology. VI (April, 1981), pp. 28-32.

Cohen, Mollie L. Educational Software: A Taste of What's Available for Social Studies. The Computing Teacher, X (December, 1982), pp. 11-15.

Cohen, Mollie L. NCSS Looks at the Computer Revolution. Social Education, XLVII (March, 1983), pp. 186-187.

Cohen, Mollie L. Computer Corner. Social Education. XLVII (October, 1983). pp. 456-459

EPIE Institute. MICROgram: Beyond Courseware. The Computing Teacher XX (January, 1983) pp. 37-38.

Fiel, Susan. Lemonade's the Name, Simulations the Game. Classroom Computer News, III (January/February 1983), pp. 34-39.

Hunter, Beverly. My Students Use Computers: Computer Literacy in the K-8 Curriculum. Reston, Virginia: Reston Publishing Company, Inc., 1984.

Hunter, Beverly. Powerful Tools for

Your Social Studies Classroom. Classroom Computer Learning IV (October, 1983), pp. 50-57.

Nilson, Jeff. The Game of Big Business. Classroom Computer News, III (January/February, 1983), pp. 32-33.

Rooze, Gene E. An Analysis of the Conceptions of Social Studies Education. Lubbock, Texas: Texas Tech University, 1976. (ERIC Document Reproduction Service No. ED 155 115).

Designing an Integrated K-12 Computer Curriculum

Barbara R. Sadowski
College of Education
University of Houston - University Park
Houston, Texas 77004

A three year district plan is presented which implements a computer-integrated curriculum emphasizing computer literacy, computer uses as a tool, and problem-solving with the computer rather than CAI. Goals are presented together with suggested topic/grade level coverage for each of the three areas. Planning, teacher-training, articulation, phased-in hardware acquisition, and formative evaluation are necessary for the successful integration of computer-enhanced instruction across grade levels.

District plans for computer curriculum implementation usually focus on the acquisition of hardware or specific objectives (See The Computing Teacher, Sept. 1981, and Jan. 1983) related to computer literacy and computer-assisted instruction. For some school districts, computers are distributed unequally across the grades and no one addresses the lack of instructional software or planning for teacher inservice in a school district. Providing all students a computer literacy curriculum during implementation is another difficult problem as is articulation between grade levels to avoid duplication of objectives. What follows is a district plan that addresses all of the above problems.

The assumptions which guided the district plan were as follows:

1. No student would graduate from the district high school without a unit on computer literacy during the 3 years of implementation.
2. District teachers should own the curricular decisions about computer objectives and activities across grades K-12.
3. Teacher inservice and training would precede implementing computer objectives at each grade level.
4. Hardware purchases would be phased in stages as teachers needed access to them.
5. All students at every grade level would have equal access to a microcomputer laboratory.

Phase I

Before deciding on goals for K-12 computer-integrated curriculum, meetings were held with district administrators and the school board. The purpose was two-fold:

1) To educate the school board and district administrators about different instructional computer uses and familiarize them with the terminology so that (2) decisions could be made about hardware and goals for the district plan.

One meeting was devoted entirely to a demonstration of software from MECC to illustrate various CAI categories. Computer literacy, programming, single-purpose tool use, and word processing were explained. Hardware and software availability and costs for each computer use were discussed along with advantages/disadvantages of each.

GOALS

After much discussion of the

merits, costs and desireability of providing tutorials and drill and practice using the computer, the following goals were adopted:

1) A useable level of skills in one programming language.

2) Ability to use computer as a tool (word processing, spread sheet, data management, graphics, etc).

3) Low level technical knowledge of how a computer works.

4) Ability to engage in problem-solving activities using a computer as a general purpose tool.

5) Experience with using two kinds of computers.

6) Perspective on social and moral issues relating to computers and large data bases.

7) Comfort level with high-technology environment and understanding of the potential of technology to improve quality of life.

Note that CAI is not a goal of this district plan. This was a deliberate decision based on the hardware costs and lack of high-quality coordinated software at this time. However, simulations and problem-solving software were included under the fourth goal of using the computer as a general purpose tool.

HARDWARE

When the goals were adopted, the district decided on a hardware purchase, formed a computer committee and developed a phased-in plan for teacher training and implementation. Although a word-processing laboratory with ten computers was available to vocational students, the decision was made to equip a laboratory in each school with 15 microcomputers, three printers and two large screen monitors. Two additional micros on carts would be provided for classroom use at each school at a cost of about $40,000/school.

A total of 32 computers would be purchased in Phase I, the first year. Seventeen would equip a laboratory at the high school for delivering the computer literacy curriculum and for teacher inservice hands-on workshops. The other fifteen micros would be divided among the remaining three schools so that teachers could practice skills learned in the workshops. In Phase II, the second set of 34 microcomputers would be distributed so that each school would have a complete lab by the end of the second year.

Software purchased included LOGO, a spread-sheet package, a data management package, a word-processing program, a drawing tool, KOALA PAD, simulations and problem-solving software.

Computer Committee

After the adoption of the goals, the school district next formed a Computer Coordinating Committee consisting of two teachers and one administrator from each school, including the teacher who directed the word processing laboratory and two district administrators. The selection of teachers for the committee was based on a survey of teacher interest, training and background in instructional computing. Although several junior and [illegible]or high teachers had taken one or more graduate courses in instructional computing, very few of the elementary teachers had any training. Responses indicated that all teachers were enthusiastic about computer inservice training, and wanted it to start immediately.

The charge to the committee in its first two months was to decide on objectives for computer literacy curriculum for the graduating seniors,

including who would teach it, and how students would be scheduled for the unit on computer literacy. In general, the committee was to decide on the objectives, scope and sequence for the computer curriculum so that students would not duplicate computer experiences as they moved through the grades.

Monthly all-day inservices were scheduled for the Computer Committee to acquaint members with district goals and to familiarize them with the hardware and software purchased by the district. At one inservice members broke into two groups, where the junior and senior high school teachers worked on the computer literacy objectives and implementation for the Spring semester while the elementary teachers continued to discuss options for delivering computer awareness inservice sessions to the other elementary faculty. The second inservice was a hands-on session using LOGO, BANK STREET WRITER and simulations from the MECC materials. The third session is planned as a hands-on workshop using the new high school microcomputer lab for a half-day, followed by planning for after-school computer inservices during the Spring semester. The expectation is that some Computer Committee members will conduct awareness and orientation inservices for other district faculty after receiving training. This is necessary because of distance and the general difficulty in enrolling in university-based computer courses. At this point, the school district has been involved in the implementation plan six months and has completed the first half of Phase I. The second half of Phase I will include implementing the computer curriculum for the high school seniors committee decisions on the objectives and implementation plan for grades 9-12, continued after school teacher inservice, and an evaluation of the curriculum at the end of Phase I. After school computer clubs for interested junior and senior high students will be started during Phase I.

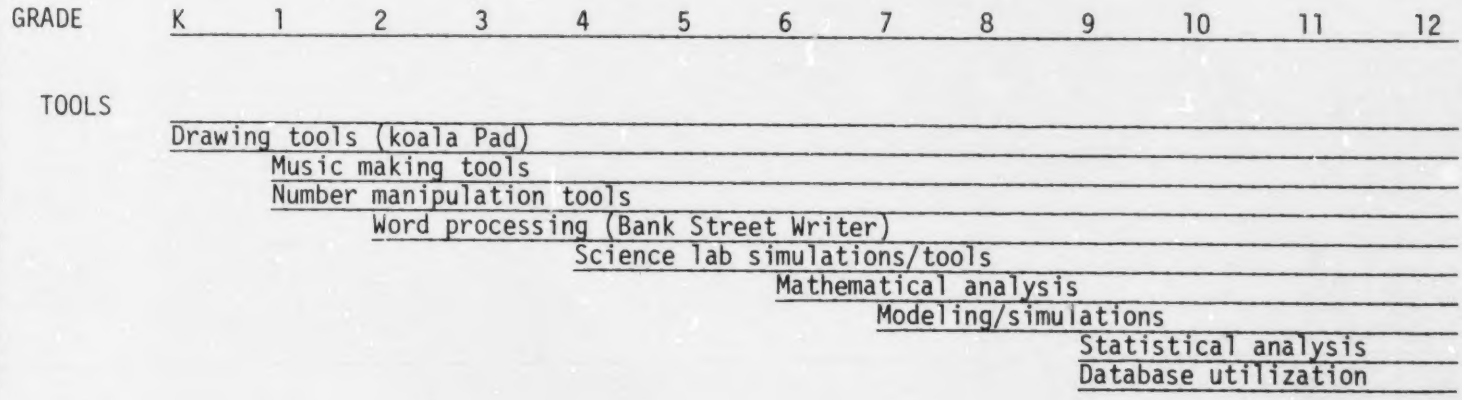
Phase II and III

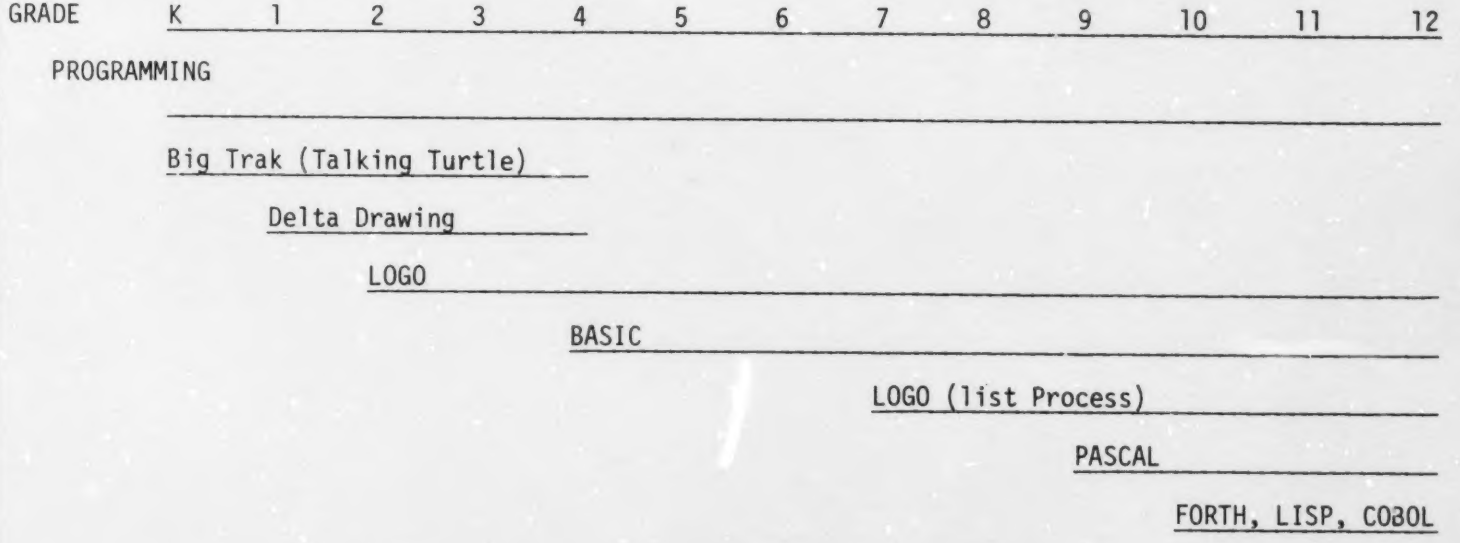
Phase II will begin with an extensive 45-hour inservice for all interested teachers. Workshops for training interested parent aides to assist in the computer lab at all grade levels will be conducted. The computer curriculum determined by the Computer Committee will be implemented for all students in grades 9-12, while the Committee works on planning the objectives and activities for grades 6-8. All these events will occur during the first half of Phase II.

Evaluation of the 9-12 curriculum will be conducted at the mid point of Phase II, followed by the implementation of the curriculum in grades 6-8, planning for grades 3-5, and continued teacher inservice for the elementary teachers.

The cycle of planning, implementation, and evaluating will repeat for grades 3-5, and K-2 during Phase III. Thus, the entire district plan will be completed by the end of the third year. Elective computer classes in PASCAL and BASIC will be offered in the high school and junior high school during Phase III. The question of implementing word processing for all students in English classes for writing assignments will be decided by the Computer Committee during Phase III in conjunction with the district administrators. Suggested Topic/Grade Level Coverage is shown in the following charts. The actual placement and specific topics are left to the discretion of teachers on the Computer Coordinating Committee. The Computer Committee will continue to revise objectives and provide articulation between schools to help provide an integrated computer curriculum across grades K-12.

TOPIC/GRADE LEVEL COVERAGE K-12





Grade K 1 2 3 4 5 6 7 8 9 10 11 12
LITERACY
Computers in our lives
Computer ethics
Computer vocational training
Business applications

REFERENCES

1. Cupertino Union School District (1983) K-8 computer literacy curriculum. The Computing Teacher, Vol. 10, #7, 7-10.

2. Fisher, G. (1983) Developing a district-wide computer-use plan. The Computing Teacher, Vol 10 #5, 52-59.

3. Kelman, P., Bardige, A., Choate, J., Hanify, G., Richards, J., Roberts, N., Walters, J. and Tornrose, M.K. (1983) Computers In Teaching Mathematics: Reading, Mass. Addison Wesley Publishing Co.

A Computer-Based Electrophysiological Laboratory for High School and College

Robert E Schaffer, Ph.D.
Microbase
8525 Stella Link Rd #5
Houston, Tx 77025

Abstract: A system uses the Apple II to sample, display, and analyze brainwaves (EEG) or heartbeats (EKG) while demonstrating the analog-to-digital conversion process. The system also includes a "game" using brainwave control to illustrate biofeedback principles. The electrophysiological laboratory offers hands-on experience in exploring fundamentals of digital signal processing relating to biological phenomena.

Introduction--a new tool for observing electrophysiological phenomena

Recently microcomputers have become powerful enough to be used in laboratory applications formerly reserved for larger computer systems. This paper describes an application of the microcomputer for the biological sciences--as a system that analyzes, displays, and stores electrophysiological signals generated by the brain or heart. As a tutorial device for demonstrating brainwaves ("EEG") and heartbeats ("EKG"), this system has advantages over prior non-computer based systems that merely amplified and displayed the biological signal. The electrophysiological laboratory's advantages include the ability to:

1) demonstrate fundamentals of analog to digital conversion techniques,
2) perform analyses on the sampled EEG and EKG, and
3) demonstrate biofeedback phenomena.

The electrophysiological laboratory is equivalent to a new form of microscope for today that allows examination, measurement, and understanding of biological processes that are not detectable without specialized equipment.

The equipment

The electrophysiological laboratory uses an Apple II computer with one disk drive, and includes a single-channel physiological amplifier, programmable timer, analog to digital convertor, as well as specialized software. The Apple II system was selected because of its high resolution graphics capabilities and the relative ease in giving the computer signal processing capabilities.

The physiological amplifier boosts the EEG or EKG signal from the microvolt or millivolt range up to several volts so that the digitizing circuitry can "read" the signal. The amplifier has adjustable low and high frequency filters that act like the tone

controls on an audio amplifier. Three electrodes, which are metal disks attached to wires, are affixed to the scalp or arms of a volunteer for recording the volunteer's EEG or EKG. These wires terminate at the input stage of the amplifier. The amplifier output is connected via cable to a peripheral card in the Apple. The amplifier is battery-powered and designed so that the user can never be exposed to the voltages found within the Apple.

The peripheral card in the Apple contains the programmable timer and the analog to digital converter ("ADC"). The timer sets the "sampling rate", which is the number of times per second the ADC converts the electrophysiological signal into a number. The timer sets this rate by generating an interrupt signal (via the Apple NMI line) at fraction of a scond intervals. The interrupt-handling software instructs the Apple to halt its present task and allow the ADC to sample the signal. A typical sampling rate for EEG or EKG would be 100 to 1000 samples a second.

The ADC converts the fluctuating voltages of the electrophysiological signal into a series of discrete numbers, with each number representing a sample of the signal. During each interrupt the ADC "takes a snapshot" of the signal and measures the signal's amplitude at that instant just as one would note the y-coordinate (height) of a line plotted on graph paper. This process is called "digitization", with one numeric sample value produced for every interrupt.

The electrophysiological signals of interest--EEG and EKG

The electroencephalogram (EEG or "brainwaves") is a continuously varying signal recorded from the scalp. The EEG is the fluctuating voltage that represents the summated activity of millions of brain neurons, although the exact relationship between brain activity and the EEG is still unknown. Over the last fifty-five years a nomenclature has been developed to charactrerize the commonly seen wave shapes. Waves are typically named according to their frequency, which is the number of times per second the EEG "wiggles" above and below its zero-volt midpoint. EEG up to 4 cycles/second is called "delta", from 4 up to 8 cycles/second is "theta", from 8 through 13 cycles/second is "alpha", and EEG over 13 cycles/second is "beta". Although certain researchers have demonstrated a relationship between a person's alpha and theta waves and the mood of the person, the only consistent relationship between these waves and the mental or emotional state of a person is the increase in alpha wave prominence if that person's eyes are closed and he or she is tired or relaxed. The typical size of the EEG is about 50 microvolts, although a "normal" range can be from several microvolts up to about 100 microvolts.

The electrocardiogram (EKG or "hearbeats") represents the electrical activity of the heart as it pumps blood. As the heart's chambers contract, a sequence of

action potentials from the heart's muscle and associated nerve tissues produce millivolt-sized rhythmically undulating electric fields that can be recorded on the body surface. Because of the relatively large size of the electric field produced by the heart, the electrodes do not have to be near the heart to record EKG. A common electrode placement uses one on each wrist or arm and a third electrode on the leg. The EKG fluctuates less than the EEG; thus the EKG nomenclature does not include a set of terms for describing a range of frequencies as with the EEG. The EKG is portrayed as a series of "waves" which are deflections from the signal's zero-volt midline. These EKG waves denote the sequence of contractions of the heart chambers. When the heart's "pacemaker" (sinoatrial node) produces an electrical signal to begin atrial contraction, a "P" wave is generated. When the electrical signal passes into the ventricles and causes their contraction, the "Q","R", and "S" waves are produced. When the ventricles return to their resting polarity a "T" wave is formed. After this wave the "U" wave occasionally occurs, although its origin and functional significance is not known.

A technique has been developed to produce a less variable EEG wave pattern for an individual. This technique involves presenting an external stimulus such as a tone or flash many times and saving about one second of EEG that occurs during each presentation. Then the pieces of the EEG from each presentation are mathematically averaged to form an average brain wave response to the stimulus. This average wave is called an "Event Related Potential" or "Averaged Evoked Potential". "Potential" in this context means "voltage". As with the EKG, typical event related potential (abbreviated "ERP") waves in respose to a light or tone have been observed and named. An "N1-P2" complex is observed starting about a hundred milliseconds after the stimulus, and a "P3" wave occurs after about three hundred milliseconds. Researchers have demonstrated variations in the size and timing of these waves according to the state of the volunteer or the information content of the stimulus.

Artifact--contamination of the signal from spurious sources

Because both EEG and EKG signals are small, they require amplification before they can be accurately measured. Unfortunately non-EEG or non-EKG signals can also be amplified, thereby distorting the signal of interest. This distortion or obliteration of the EEG or EKG is referred to as "artifact", and is the nemesis of any electrophysiological recording system. Some causes of EEG artifact include eyeblinks, muscle tension, sweating, swaying electrodes, or even EKG. Muscle tension also distorts the EKG; both EEG and EKG will be distorted by improperly applied electrodes or faulty instrumentation. A major purpose of the electrophysiology laboratory is to allow the student practice in identifying and eliminating the source of artifact in a contaminated EEG or EKG

signal.

The electrophysiology laboratory software

Two sets of program modules have been developed that demonstrate the traditional brainwave analysis techniques: one displays ongoing EEG, the other performs an experiment that displays event related potentials. An electrocardiogram module displays and analyzes ongoing EKG. A final program demonstrates EEG biofeedback principles in a game context.

The EEG oscilloscope program module

This module samples and displays one channel of spontaneous EEG activity on a television monitor using high resolution graphics. The program operator can alter the horizontal point resolution and can select the sampling rate to be between 20 and 800 samples/second. Adjusting these parameters expands or contracts the visual display to make certain features of the EEG more recognizable. Sampling can be suspended to "freeze" the screen for careful examination of the displayed EEG patterns. This display can then be saved as a disk file for future inspection. The display can also be analyzed with the module's wave measuring option that underlines each theta and alpha wave. The module also contains several demonstration EEG files that show commonly-observed phenomena including an alpha burst and eyeblink or muscle tension artifact.

The event related potential program module

This program module presents a visual stimulus every few seconds and saves about one second of EEG associated with each presentation. The program then averages the saved EEG to form a grand average ERP representing the typical response to the stimulus. The visual stimulus consists of two faces, each with five features (eyes, nose, mouth, hat style, and tie style). The faces are identical one-quarter of the time, the other presentations show faces differing by one randomly-chosen feature. After each presentation the participant must press a button to denote whether the faces were identical. The program creates two ERPs--one for the typical stimulus condition (different faces) and one for the rare stimulus condition (identical faces). The ERP to the rare stimulus should have an enhanced "P3" wave in comparison to the typical stimulus ERP. This amplitude enhancement seems related to the person's analyzing the information content of the rare stimulus. These ERPs are displayed in high resolution graphics on the t.v. monitor, and can be saved in a labelled text file for subsequent viewing.

The electrocardiogram program module

This module samples and displays the EKG in high resolution graphics on the t.v. monitor. The number of points displayed across the screen and the sampling rate can be

altered on-line to accentuate certain visually-detectable waves. For example, the module can squeeze two heartbeats onto the screen or can give an expanded view of just one heartbeat. The display can be suspended to examine a particular heartbeat sequence. One option lets the operator measure the time interval between any two points on the screen--enabling measurement of the heartbeat rate or the relative timing of the P, Q, R, S, and T waves in a single heartbeat. The displayed EKG can be saved as a file on disk for later examination. Other options display simulated EKG screens or EKG demonstration screens.

The EEG biofeedback program

This program demonstrates the principle of biofeedback--one can develop a degree of control over certain bodily processes such as brainwaves, blood pressure, or heartbeat rate by monitoring changes in the specific process you are trying to control. A typical EEG biofeedback device generates a tone or light whenever the participant generates an above-threshold amount of alpha activity. The participant is instructed to develop mental strategies to keep the light or tone on. Often the threshold has to be manually readjusted if the person starts to tire or if the person's proficiency increases.

This program displays one second of EEG on the t.v. monitor, estimates the amount of alpha activity in that second, and changes the pitch of an emitted tone and the position of a rocket shape on the screen based on the alpha estimate. If the participant produces a relatively large amount of alpha the tone is high-pitched and the rocket moves up the screen. Low amounts of alpha produce a lower-pitched tone and moves the rocket down the screen. The threshold for deciding whether the most recent second's worth of alpha is "large" or "small" is based on the average alpha estimate of the most recent four seconds of EEG.

There are several advantages of this biofeedback program over the traditional biofeedback systems:

1) the rocket and tone are more interesting to observe and manipulate than the modulated tone or winking light typically used,

2) the EEG is displayed so that artifact can be detected and eliminated rather than causing a "false feedback" situation to non-EEG signals, and

3) the alpha activity threshold is "adaptive" to the participant's performance level. As the person tires or gets more adept at generating alpha, the alpha threshold alters. Thus the task never gets too hard or too easy.

This biofeedback program can also be considered a rudimentary biocybernetic training device that allows the participant the opportunity to develop his or her ability to control a graphics display via EEG.

Summary--the value of the electrophysiological laboratory as a teaching tool

The electrophysiological laboratory provides a set of unique learning experiences that extend beyond the biological sciences. It offers a "hands on" environment for understanding the fundamentals of digital signal processing by allowing on-line manipulation of the sampling rate, the signal amplitude, and the high and low frequency filters. The EEG and EKG program modules can perform elementary wave analyses on the sample of data displayed, and can archive the display as a disk file. The event related potential program module performs an experiment that elicits a well-known electrophysiological effect showing how the information content of a stimulus can alter the brain's electrical response. The EEG biofeedback program illustrates the principles of biofeedback using a task that is inherently more interesting than traditional biofeedback displays.

The electrophysiology laboratory can thus be used to demonstrate signal processing fundamentals to physics or electronics students, biofeedback principles to psychology students, and give valuable experience in recording and analyzing EEG and EKG for the biology or health science students. What is most important is that the electrophysiology laboratory allows the student to learn to utilize the computer in a real-world digital signal processing application.

Planning Your School Computer Program

Susan K. Sclafani
Implementation Coordinator
Department of Technology
Houston Independent School District
5300 San Felipe, Houston, Texas 77056

Before implementing a computer program at the school level, the staff should follow a systematic planning process, including defining the instructional need, identifying the appropriate applications, determining the location of the computer equipment, identifying the students to be served and the teachers to be trained and selecting the hardware required. Each of these steps includes important questions to be considered. Particular attention should be given to the issue of determining which applications: tutor, tool, tutee will be offered to specific subsets of the student body: minority students, females, remedial students and gifted students.

Microcomputers offer educators a new and exciting resource to be used in the classroom. It is, however, important to engage in a careful planning process before investing scarce school funds in computer hardware and software. The first step in the planning process should be to define the instructional need. If that need can be met as well through worksheets and textbooks, there is little justification for the use of computers. The use of "worksheets on the screen" does not take advantage of the capabilities of the computer, nor does it teach students why computers are considered such valuable tools. Instead, the needs should be those that are not now being satisfactorily met by the resources available to teachers.

It is important to state the instructional needs as specific measurable objectives so that the staff is clear what they expect to accomplish and can evaluate the progress toward that objective. Examples might be:

Students scoring below the 50%ile in Arithmetic Computation on the ITBS will improve their computational skills through the use of drill and practice and tutorial programs as measured by their performance on the ITBS in March. It is to be noted that one cannot predict specific measures of growth at this time because of the lack of definitive research in this area. The careful compilation of data from the school's program should be analyzed to determine the gains and make better predictions the following year. The publication of such studies would enhance the knowledge of educators across the country.

The next step in the planning process is to identify the applications and the specific software packages that will meet the instructional needs defined. The staff should study the applications available. Robert Taylor's designation of the computer as tutor, tool and tutee is a helpful designation. The computer as tutor includes what is usually thought of as

computer assisted instruction. Drill and practice programs assist students in maintaining and reinforcing skills. Tutorial programs teach concepts by guiding students through a sequence of steps. Simulations model situations that are impossible or impractical for the classroom. Problem-solving programs include logic exercises for younger students and mazes and puzzles for older students. Instructional games embed skills objectives in a game format. In all of these applications, the computer is in control of the interaction and is therefore in the role of the tutor.

The computer as tool includes both wordprocessing and data management. These uses are transferable to the office as well as to professional careers and should be introduced to all students. Wordprocessing can take the drudgery out of writing and revising and thus encourage students to improve their writing skills.

The computer as tutee means that the user is in control of the interactions with the computer. Programming is the prime example. As Seymour Papert points out in Mindstorms, students who are taught to program learn the concept of debugging, which can be applied to any algorithms. Thus a student who has difficulty dividing can learn to "debug" his division algorithm, rather than decide that he cannot divide. This feeling of control over his learning can lead to the overcoming of past barriers. That is one reason why all students should be taught the basics of programming. A second and equally important reason is that this allows the student to truly understand the power of computers, rather than feel intimidated by the mystery of how computers work.

The selection of specific software packages is a topic worthy of its own paper. It is important to research those packages that have received positive reviews and then review them with an eye to how well they meet the defined instructional needs of the school's students.

The third step is to determine the location for the computer equipment. The considerations here are threefold. Security must be attended to as few schools can afford insurance or replacement. Rooms with easy access through the windows should be avoided. Doors should be solid and have separately keyed locks. Rooms opened by the custodian at 7:00 AM can be empied by the time the teacher arrives at 7:30.If the school has an alarm system, a sensor should be placed inside the room to apprehend anyone, who has made it past the door and window precautions.

A second consideration is the expertise of the teachers. If there are only one or two teachers comfortable with using the computers, the best decision would be to place all of the computers in a single location. Teachers can then be inserviced and allowed to observe the computer teachers in the lab. If all teachers are ready to use the equipment, the lab might still be the best choice. Teachers could schedule their students into the lab when they were going to use the computers and leave the lab to other teachers, when the students were working on other activities. Teachers, who use

a learning center approach in their classroom, can easily add the computer as another station. This decision should be made by the staff, so that no teacher feels excluded from computer use.

The third consideration is the electrical requirements for the equipment and the money available to make it adequate. A microcomputer uses two to four amps of power and requires two three-pronged plugs. As the number of computers increases, the likelihood of having sufficient outlets existing in a room decreases. Consideration should be given to old typing or homemaking rooms, which have already been wired. The cost of adding outlets can range from $300 to $2000, depending upon the number of outlets and the amount of power already coming into the room.

The fourth step in the planning process is to determine the students, who will receive computer instruction. There are a number of ways to do this. If the cited need specified a sufficiently small number of students via a vis the number of computers, the task is easier. One must still ask how the students will be scheduled. Will the students be "pulled out" of regular classes or will whole classes be scheduled at the same time? How will teachers accomodate themselves to this schedule? It is important to plan the program so that all students will have access to the computers at some point during the year. It would be a mistake to identify the computers as the exclusive domain of any group: as a remediation lab or a gifted student activity or a male domain. Students are very quick to perceive how the school has identified the appropriate groups to use the computers. A second problem is one that has been noted in the literature; poor students are introduced to drill and practice while gifted students are taught to program. The staff should plan to avoid this problem by building access to all application types for all students into the program. This may be accomplished by reserving one day per week for teachers to sign up for the computer lab or for one computer to use with their classes. This can be done on a rotating basis so that all students understand that they will have a chance to use the computer as an individual or as a class. A committee of teachers can be formed to supervise this scheduling and to insure that the computers are always used.

Once the staff has completed the steps listed, it is ready to choose the hardware. This choice will be dictated by the decisions already made. The "right" computer will be the best one the school can afford that has the software available to meet the instructional needs. Given that 30 students would receive an hour of computer time per week on one computer, the number of computers can be figured as a minimum and a maximum to work toward. In a large school, the minimum might be figured on allowing each class an hour a week at the beginning. There are many excellent programs that can be used by groups of students from two to thirty. Choices must also be made about the need for green or color monitors. Elementary school programs often make excellent use of color as a motivational tool as well as a graphics tool for delineating parts of a

diagram. For programming, word-processing or progress reports from a management system on a program, a printer is another need. Fortunately the price of printers has decreased to the point of making them an affordable addition to the computer setup. A committee of staff members should visit a number of different computer stores to determine the best prices, the best service available at the best price for maintaining the equipment and the possibilities for staff training from the vendor. A comparison matrix will illustrate the best choice and will enable the committee to justify its choice to the staff and the administration.

The last step is to arrange for the training of the teachers. If there are one or two teachers who are already knowledgeable, they can set up a series of inservices for the rest of the staff. If there are none, the staff can arrange for an inservice through the vendor, a local school or a university teacher. Teachers should be encouraged in this case to take courses at the local university so that the staff's knowledge can be continually upgraded. There should be some teachers with a knowledge of programming, although it is not necessary that every teacher know how to program. The staff can build on the strengths of its teachers by scheduling students for different activities with different teachers.

This planning process may seem like a long one, but the end product will be a computer program that is exciting to all of the students and teachers and one that will grow with the knowledge and experience of of the staff and the students. Students may grow beyond the expertise of their teachers and they should be encouraged to contribute this knowledge to their fellow students and teachers. Students as well as parents make wonderful assistants to the busy teacher and should be utilized in this capacity. A successful program is a growth experience for all involved and will add more to the school than just the acquisition of computer expertise.

IMPLEMENTATION OF THE LITTLE LOGO CURRICULUM IN THE FOURTH GRADE

Dr. Pat Semmes, Captain Jackie DeWolfe, Mr. Paul Rode, Miss Jill Caldwell, and Dr. Kay Dillingham
c.o Cambridge Elementary School
1001 Townsend
San Antonio, Texas 78209

LITTLE LOGO, a programming language, has been developed for children. This language converts turtle graphics to a single-keystroke activity and serves as a transition to the Logo language. Accompanying activities emphasize higher level thinking skills related to computer literacy and mathematics. A pilot project was implemented at Cambridge Elementary School, Alamo Heights Independent School District, in the fourth grade.

Logo is the language most often recommended by educators for study in the elementary grades, especially because of its graphics capabilities. The Logo graphics turtle can be taught (i.e. programmed) to draw geometric shapes. In order to have the turtle draw a straight line a command such as FD 30 must be typed, followed by pressing the return key. This involves six keystrokes and is difficult for a young child unfamiliar with the keyboard. Although classroom teachers may know the keyboard, they may not know the syntax of the language or how to operate the computer hardware. Many teachers who have no computer background are being required to teach their students while learning themselves. This contributes to teacher frustration and computer under-utilization.

LITTLE LOGO is a single keystroking programming language which was developed to overcome some of the complexities involved in using Logo and to provide a transition to Logo for students and teachers. For example, pressing "F" replaced the command "FD 20" in Logo. The language is actually a series of programs presented by increasing levels of difficulty. Each level generally contains the commands of the previous level as well as a set of new commands. In LITTLE LOGO the commands are suggestive of the commands of Logo. Activities to accompany the program levels have been developed. In the activities the children act as programmers working on geometry related graphics programs.

Many states either require or will soon require computer literacy to be included in the curriculum. Also, there has been increasing parent awareness of the need for their children to be computer literate in order to compete in todays society. Being aware of this, Mr. Paul Rode, the principal of Cambridge Elementary School in the Alamo Heights Independent School District, attended Logo Users Group meetings conducted by Dr. Pat Semmes at Trinity University. He was later asked to help evaluate the LITTLE LOGO curriculum during its development. After discussion it was decided that Cambridge would take part in the pilot program to test the LITTLE LOGO curriculum. Mr. Rode wrote a proposal to the school district which was accepted. He then purchased four Commodore 64 computer systems, one for each fourth grade classroom.

One of the greatest problems in implementing computer programming is the lack of teacher knowledge about computing. Researchers have found that most teachers do not feel at ease in the presence of computers and that teachers' attitudes and expertise effect the successful implementation of the curriculum. At Cambridge care was taked to insure that all of the fourth grade teachers were adequately instructed in the use of the hardware and sortware involved in the LITTLE LOGO curriculum. This was accomplished by the means of several training sessions, as well as by trainers being available during the initial hands-on group exposure of the children. Training was provided by Dr. Pat Semmes and graduate students from Trinity University, Jackie DeWolfe and Kay Dillingham. The school district also provided for the children a consultant teacher who taught them an introduction to computers.

Jill Caldwell, fourth grade coordinator, and Mr. Rode felt that all teachers should practice similar classroom management techniques when implementing the LITTLE LOGO curriculum. After some experience however, various approaches were tried. Each fourth grade classroom had its own computer system and each teacher was permitted to make changes to best suit her classroom schedule. After consulting with the children, some teachers felt pairing of the students was best while others preferred each student to work alone. Some teachers wanted the children to work on the computer for ten minutes twice a week while others preferred that their students work twenty minutes once a week in order to avoid conflicts with the rest of the curriculum.

This pilot program is still in progress and final results will be forth coming. So far response has been gererally positive from the teachers and students. At the beginning of this experience, teachers expressed anxiety and wanted the trainers to be present when they first introduced the curriculum to the classes. Their fears proved to be well founded when a disk drive malfunctioned and the magnets in a classroom upstairs kept crashing the program. Currently, calls to the trainers for help seem to have ceased. Apparently, LITTLE LOGO has served its purpose since some of the fourth grade teachers are investigating the possibility of moving on to Logo. If this occurrs, the teachers have projected that the LITTLE LOGO curriculum might best be taught in the third grade so that they may teach Logo in the fourth grade.

THE EGG COMPUTER: A SIMPLE MODEL

Pat Semmes
Computing and Information Sciences
Trinity University
715 Stadium Drive
San Antonio, Texas 78284
(512) 736-7366

The EGG series computer as described by Statz and Miller (1978) is an easily constructed model of a computer which can be made from one or more egg cartons. Using this model, students can trace the execution of programs whose statements consist of a subset of the BASIC language. It's use is appropriate for students just beginning a study of BASIC as well as more advanced students who have an interest in the computer's organization.

Students tend to view the computer as a magical black box. The lack of understanding of what is happening inside the machine often leads to difficulties. For example, statements involving variables such as LET X = X + 1 typically causes problems. To alleviate such problems, students can construct a simple model of a computer and trace the execution of programs using the model. For this purpose the EGG computer or, more specifically, the EGG12 computer can be used. It is especially helpful for students studying the BASIC programming language.

The EGG12 computer actually consists of the following:

1. an EGG CARTON which serves as a computer memory of twelve memory cells
2. STRIPS OF PAPER for the program statements that are placed in memory cells
3. an ARROW which points to the memory cell containing the program statement being executed
4. ADHESIVE LABELS for variable names that are attached to memory cells
5. PAPER SQUARES which contain data to be placed in memory cells during execution of program
6. a BLANK PIECE OF PAPER which acts as the computer screen

Sample sizes for the materials needed other than the egg carton and blank piece of paper are shown in figure 1. Sizes can be changed, if desired.

Students write program statements onto the program strips, one statement per strip. The line numbers begin with one and are numbered consecutively. The memory cells in the egg carton are numbered from one to twelve, and the program strips are placed into the corresponding cells. The variable names are then written onto adhesive labels (one label per name) which are stuck onto empty cells. These cells are then ready for storing data. As the program is executed, the arrow serves as the program counter and is moved from cell to cell so that is always points to the memory cell containing the instruction currently being executed.

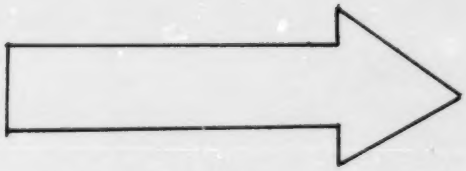
The following statements from the BASIC programming language are used: PRINT, LET, INPUT, GOTO, IF, and END. A brief description of each statement follows:

PRINT - The PRINT statement has the usual meaning. Both the values of variables and messages within quotes

PAPERSTRIPS FOR PROGRAM STATEMENTS

PAPER SQUARES FOR DATA

LABELS FOR VARIABLES



ARROW (colored)

Fig. 1. Full size models

can be printed. Note that printing occurs on the sheet of paper which serves as the computer screen.

LET - For the LET statement the values for any variables on the right side of the equal sign are obtained by examining appropriate memory cells, any necessary computations are carried out, and the result is written on a paper square and placed in the memory cell named by the variable on the left side of the equal sign.

INPUT - When the INPUT statement is executed, a ? is placed on the "computer screen." The user then writes the number to be input on a paper square and places it in the memory location corresponding to the variable name in the statement. The number is also written on the "computer screen" next to ? .

GOTO - The GOTO statement causes the arrow to be moved to the memory cell with the number indicated in the statement. The instruction in this memory cell will be the next instruction executed.

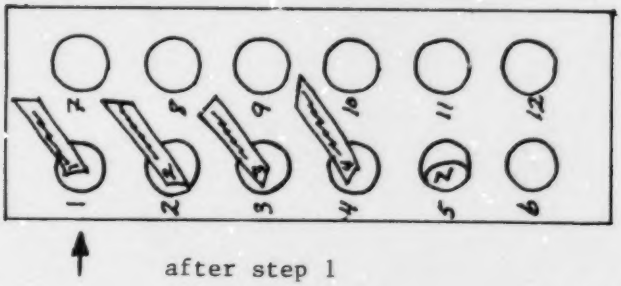
IF - When executing an IF statement, if the condition to be tested is true, the arrow is moved as in the GOTO statement. Otherwise, the arrow will be placed at the next memory location.

END - stops execution of the program.

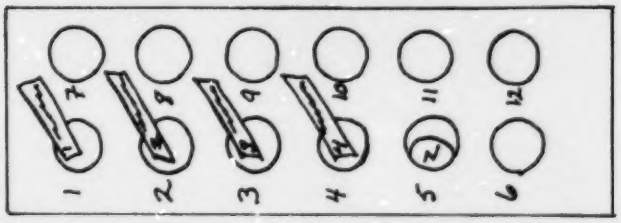
```
1 PRINT "WHAT IS YOUR FAVORITE NUMBER?"
2 INPUT N
3 PRINT N "CHEERS FOR TCEA"
4 END
```

Fig. 2. Sample Program

Program Ready For Execution

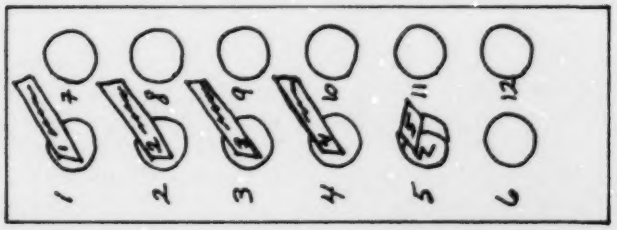


after step 1



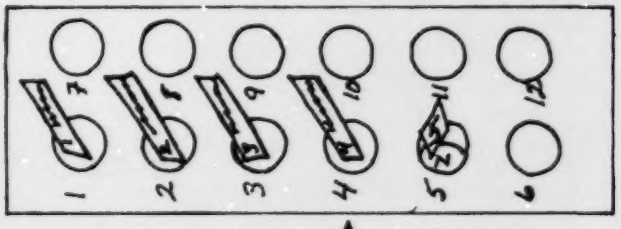
WHAT IS YOUR
FAVORITE NUMBER?

after step 2



WHAT IS YOUR
FAVORITE NUMBER?
?5

after step 3



WHAT IS YOUR
FAVORITE NUMBER?
?5
5 CHEERS FOR TCEA

after step 4
has been removed

Fig. 3. Execution of Program

In order to run the sample program shown in figure 2, the student first clears the memory cells of program strips, data squares, and various labels from the previous program. The statements are then copied onto four program strips which are placed in the first four memory cells. The variable name N is then written on an adhesive label which is stuck onto memory cell 5. The arrow (program counter) is then placed so it points to the first memory cell as shown in the top diagram in figure 3.

In order to execute the program, the student locates the memory cell indicated by the program counter and "fetches" the instruction from the memory cell. He "executes" the instruction after returning it to the memory cell and then "increments the program counter" by moving the arrow to the next memory cell. (In the case of the GOTO or IF statement the arrow may be moved to a different cell.) This process, referred to as the "fetch execute cycle" is performed over and over again until the END statement is executed. The execution of the sample program is illustrated in figure 3.

The execution of other more difficult programs can be demonstrated using the Egg Computer Series. Using this model, students will gain valuable insights into the organization of computers which will be helpful in tracing programs and understanding the components of a computer system. For a description of more advanced programs and concepts relating to the Egg Computer Series, the reader is referred to Statz and Miller (1978).

Reference:

Statz, Joyce and Miller, Leland, "THE EGG SERIES: USING SIMPLE COMPUTER MODELS," The Mathematics Teacher , May, 1978.

MATH DOCTOR, M.D. MICROCOMPUTER ADAPTIVE DIAGNOSIS

Dr. Barbara Signer
Curriculum and Instruction
College of Education
University of Houston
Houston, Texas 77004

The session will consist of a hands-on demonstration. Participants will be diagnosed for arithmetic skills using the program MATH DOCTOR, M.D.

Microcomputers have been successfully used as a simulator for professional training, especially in the field of medicine. Why not use microcomputers to simulate a mathematics diagnostician? This computer diagnostician could have patience beyond human capabilities, accept generative responses (improving upon machine scored tests), provide immediate diagnosis upon completion of the test and tailor the test according to the user's responses.

Computer Adaptive Testing - CAT

There have been a number of studies of Computer Adaptive Testing (CAT). Among the aspects of CAT that have been studied are reliability and validity, accuracy at extremes, information yield, effects of varying step sizes, measurement errors and the number of items presented.

The basis of CAT is the utilization of a branching technique that analyzes a student's response to select the next appropriate test item. Various algorithms are contained in the psychometric literature. CAT branching research had its beginnings in a Princeton based statistical research group. It focused on sequential analysis and a "staircase" testing method in the early 1940's. Psychometric origins of adaptive testing can be traced back to the 50's to the work of Hick who described a branching process in which successive items would be chosen in such a way that each would have a 50% chance of being correctly answered by the subject, based on the answers to the previous items (Thorman, 1976).

Deihl and Rodway, as cited in Lippey (1974), devised a computer adaptive "incremental test" in which test items were arranged linearly by degree of difficulty. Their study occurred in 1974. If a correct response was given, it jumped items, if not it jumped back, moving in a linear manner through 20 items in the test. There were three ways for the test to terminate; a student reached the end of the test, a student made three errors in succession, or a student made a total of five errors.

Epstein, as cited in Holtzman (1976), devised an item pool test strategy in 1973. If given statistical specifications (ie. acceptable range of means, acceptable range of standard deviation of difficulty, and number of items needed for each category), the computer surveys an entire item pool, produces a reduced item pool and assigns a priority number to each item.

Robbins and Monro, as reported by Lippey (1974), used a Stochastic approximation process. A harder item is administered after each correct answer and an easier item after each wrong answer. Starting with large changes in item difficulty it then uses successively smaller steps to zero in on the level of difficulty at which the student answers correctly 50% of the time. This difficulty level is a measure of the student's standing on the trait measured by the test.

The Educational Testing Service (ETS) Broad Range Tailored Test of Verbal Ability (BRTTVA), developed by Lord and cited by Stocking and Swanson (1980), uses only 25 items. It computes an estimate of a student's ability using responses to all previous items. It then determines the next type of item to be given. The test can terminate without administrating all 25 items.

Microcomputer Adaptive Diagnosis - MAD

The studies cited in the previous section were conducted on main frame computers which selected questions from existing test item banks. With the increasing availability of microcomputers in the home and school it is logical to extend CAT to the microcomputer. The following capabilities of the microcomputer make it a desirable diagnostician: extended memory, random generating function, the ability to interact with the user, the ability to branch and immediate scoring and analysis of results.

A Model of Microcomputer Adaptive Diagnosis

In response to the appeal of the microcomputer as a diagnostic instrument the program MATH DOCTOR M.D. was written. The intention of the developer was to have it serve as a model for microcomputer adaptive diagnosis.

The first step in creating the model was to identify a learning skill hierarchy. An arithmetic hierarchy was determined after examining other arithmetic diagnostic tests (ie. Stanford Diagnostic Mathematics Test and Key Math Test). It was then refined to be consistent with the model hypothesized by Uprichard (1980) and with the scope and sequence of elementary mathematics textbooks.

The skills in the hierarchy are catagorized as follows:

I. Number Concepts
   - A. Whole Number Sequences
   - B. Whole Number Rounding
   - C. Rational Notation

II. Computation
   - A. Whole Numbers
     1. Addition with and without renaming
     2. Subtraction with and without renaming
     3. Multiplication of basic facts through multidigit factors
     4. Division with single and multidigit divisors and dividends
   - B. Rational Numbers
     1. Addition of proper fractions and mixed numerals with and without like denominators
     2. Multiplication of proper fractions and mixed numerals
     3. Division of proper fractions
     4. Subtraction of mixed numerals

The branching logic was formulated by sequencing objectives within each group. If the youngster misses an objective that is a prerequisite skill for the following objectives within that grouping the program jumps to the next category. All answers are generative, that is no choices are given. The use of the random generating function to construct examples allows for multiple examples to test any given skill. Upon completion of the test, the length of which varies according to the user's

responses, the results are printed on a screen, a printer, or saved on a disk. The output shows specifically where the student is first experiencing difficulties within and across each of the categories. By consulting the accompanying skill learning hierarchy, the parent or diagnostician can determine those skills which are within the student's arithmetic repertoire.

Preliminary Study

An exploratory study was conducted to gather information on the ability of the program to diagnose youngsters. Three students, grades 4-6, participated. Their selection was based on their enrollment in the Diagnostic Learning Center. Each child was diagnosed by the MATH DOCTOR program and by a pair of graduate students who followed the protocals of the Maryland Diagnostic Model (Wilson and Sadowski, 1976). The results of the microcomputer diagnosis and clinician diagnosis can be obtained by writing to the author. An examination of the results revealed high agreement between the two diagnoses. No attempt is made to claim this demonstrates reliability of the microcomputer tests or that the present program and its hierarchy are beyond modification. The author only offers the results of an exploratory study to consider future structured testing.

Future Study

It is suggested that this model be used with other disciplines in which a learning hierarchy can be identified. Branching strategies could then be mapped according to prerequisite skills similar to those used in the MATH DOCTOR program.

In addition future research needs to conducted on interpretation of microcomputer diagnostic results using this model. The results are not norm referenced as are results of standardized tests. The results of this model are closer to being diagnostic in that they take a task analysis approach, relating the learner's performance to a hierarchy of skills, rather than quantitatively to a group.

It is important to note that the author does not assert that the test and its hierarchy should not be considered for modification. In fact, the author welcomes suggestions. However, the author offers her model for; test construction, branching, use of microcomputer capabilities and interpretation.

REFERENCES

Edwards, J.S. (1980). Computer-assisted testing. Journal of Research and Development in Education. 14 (1), 9-15.

Holtzman, E. (1970). Computer assisted instruction-testing and guidance. New York: Harper and Row

Krettzberg, C.B., Stodking, M.L., and Swanson, L. (1977) Computerize adaptive testing principles and directions. Computers and Education. 2 (1), 319-329.

Lippey, G. (1974). Computer-assisted text construction. Englewood Cliffs: Educational Technology.

McKinley, R.L. and Reckase, M.D. (1980). Computer applications to ability testing. AEDS Journal. 13 (3), 193-203.

Rudner, L.M. (1978). A Short and Simple Introduction to Tailored Testing. Paper presented at the Annual Meeting of the Eastern Educational Research Association, Williamsburg, VA.

Signer, B.R. (1982). Math doctor, M.D. The Computing Teacher. 10 (5), 16-21.

Stocking, M.L. and Swanson, L. (1980). Computerized Adaptive Testing. The Computing Teacher. 8 (7), 12.

Thorman, J. (1976). Computer assisted testing: An alternative assessment option for students. Computers and Education. 3 (1), 207-209.

Uprichard, E., Underhill, R., and Heddens, J. (1980). Diagnosing mathematical difficulties. Columbus: Merrill.

Urry, V.W. (1977). Tailored testing: a successful application of latent trait theory. Journal of Educational Measurement. 14 (2), 181-186.

Wilson, J. and Sadowski, B. (1976). Maryland Diagnostic Test, University of Maryland.

Computer Literacy for Teachers

Dr. Barbara Signer
Curriculum and Instruction
College of Education
University of Houston
Houston, Texas 77004

The focus of this paper is on the responses of colleges and universities to meet the needs of teachers to acquire computer literacy skills. A detailed description of the course offering at the University of Houston is included, where the author is a member of the faculty.

State departments of education have announced that computer literacy skills will be required of youngsters throughout grade levels. School districts around the country are instituting computer literacy courses or including components in existing courses. Who will train the teachers? The focus of this paper is on the responses of colleges and universities and in particular the College of Education at the University of Houston, where the author is a faculty member.

RESPONSES FROM COLLEGES AND UNIVERSITIES

An examination of the objectives of pre and inservice courses include the following:

1. An understanding of a microcomputer's component parts
2. The knowledge of what microcomputers can and cannot do in a computer based educational system
3. The identification of the impact microcomputers will have on present and future education
4. Information on the availability of inexpensive educational resources
5. The opportunity to examine software that is available for instructional and management uses
6. Training on how to evaluate instructional/managerial programs
7. Training on how to modify instructional/managerial programs
8. Instruction on how to write instructional/managerial programs
9. Training on how to use the microcomputer as a tool
   a) word processor
   b) information management
10. Instruction on how to use the microcomputer as a tutee (LOGO)
11. Instruction on how to use microcomputers as aids in problem solving

12. A knowledge of the effects of computers on individuals and society

13. A knowledge of the history of computers

EXAMPLES OF COMPUTER LITERACY COURSES FOR EDUCATORS

Many colleges and universities offer computer literacy courses in various departments, reflecting different student populations. Descriptions of computer literacy courses that meet the needs of teachers are contained below.

Pepperdine University

Coping with Computers: Topics covered include component parts, functions, history and future of computers; how to write simple computer programs, use word processors and play computer games ("Computing and," 1983).

University of Oregon

Introduction to Computers in Education: Objectives include acquiring knowledge and skills related to teaching using computers, teaching about computers as a part of one's discipline area and dealing with change in the content of the current curriculum to reflect the capabilities of computers. One third of the course would be hands-on experience with programming in BASIC, LOGO and PILOT (D. Moursund, personal communication, May 1981).

University of Denver

Computer Literacy for Teachers: Topics include definitions of technology, computers, computer languages, computer programming; discussions of the interface between technology and society, artificial intelligence, home and office of the future, computer careers, simulations and the future of the micro millenium.

Developing CAI Units for the Microcomputer: Designed for teachers to develop software in PILOT, FORTH, BASIC and other authoring languages (R. Hoffman, personal communication, October 1982).

University of South Florida

Microcomputers in Education: Advanced programming in BASIC, selection and evaluation of software and hardware, applications of computers in education, types of CAI, networking, and computing resources.

LOGO: Methods and Applications: A study of LOGO philosophy; programming in LOGO: methods and materials for teaching LOGO (W.C. Engel, personal communication, April 1983).

Texas Tech University

Computing Literacy for Elementary Teachers: Topics include: using a calculator, operating a computer and programming in BASIC (Anderson 1983).

COMPUTER LITERACY AT THE UNIVERSITY OF HOUSTON

Computers in the Classroom

While trying to offer a wide variety of topics and training, consideration was given to the vast amount of hand-on time required for a quality course. For this reason three areas of emphasis were selected. They are a) computers in a problem solving environment, b) teacher authoring languages and computer literacy curricula.

Computers in a Problem Solving Environment

There has been a change in the teacher population requesting computer courses. More elementary and primary school teachers are signing up for these courses than in previous years, when secondary mathematics teacher dominated the classes. Microcomputers have moved into the early grades with computer assisted instruction (CAI) and LOGO. As a computer language, LOGO creates a problem solving environment in which the child teaches the computer. The child learns to

experiment in the LOGO world and refine and structure his/her thinking. Unlike CAI teachers need specific training in the LOGO language. The literature is full of reports documenting the rapid pace in which youngsters in the early grades progress with LOGO. Teachers need confidence to guide these eager minds that have no limit to what they absorb. Only with extensive hands-on experience, simulating the child's (through experimenting with peers) can a teacher introduce and lead youngster's through LOGO.

Teacher Authoring Languages

Until recently teachers were writing educational software exclusively in BASIC. While a popular language for beginning programmers, it was not designed for creating instructional materials. However, PILOT as a language lends itself to this purpose. The introduction of SUPERPILOT makes PILOT even more powerful for teachers. Graphics and sound become even easier and are important for CAI.

Computer Literacy Curricula

Many principals are new to computers in education and turn to their faculties for assistance. Often teachers have only had a single course in BASIC programming. For this reason a study and design of a computer literacy curricula, is included. The teachers first examine and evaluate available software and then investigate the resources of their districts. With this knowledge and what appears in recent computer journals, a teacher designs a computer literacy curricula appropriate for his/her level and area of teaching. The report is done on a word processing package appropriate for schools.

Conclusion

With the increased computer instruction in the schools, the impact on the individual and new developments to come, the content of computer literacy courses for educators should change. Today's youngsters that will become teachers will enter higher education with computer skills. Colleges of education will not have to play catch up with the surrounding schools. These candidates will be using computers as tools (word processors and data management). With this in mind there will be less need for general computer literacy, but more need for training on specific new developments in hardware and software.

REFERENCES

Anderson, R. (1983, March). Computer literacy for elementary teachers. Paper presented at the meeting of the Texas Com puter Education Association, Austin, TX.

Computing and higher education: The revolution is through the gates. (1983, May/June). The Forum for Liberal Education. pp. 2-18.

CONTROLLING THE COMPUTER DEMONSTRATION PROCESS

RICHARD ALAN SMITH, Ph.D.
TECHNOLOGY PROCUREMENT COORDINATOR
DEPARTMENT OF TECHNOLOGY
HOUSTON INDEPENDENT SCHOOL DISTRICT
5300 SAN FELIPE
HOUSTON, TEXAS 77056

ABSTRACT

A set of procedures is presented which will allow purchasers to evaluate, on their own terms, computer products demonstrated by vendors.

The demonstration of all brands of computers under consideration for use by a school district is one of the most important events of the purchasing process. This is true whether you are buying one computer for a classroom or hundreds of computers for your entire school district.

It is at the demonstration that you will have an opportunity to learn if the computer that you are evaluating can really do what its advertisements say it can. Also, and more importantly, you can see which systems can efficiently run the software you need to use in order to meet your defined needs.

Getting maximum benefit from demonstrations requires that you know what features you need a computer to have and that you ask to see those features demonstrated.

HOW TO KNOW WHAT FEATURES YOU WANT TO SEE DEMONSTRATED

Identify the Needs of Your Users

Before you get to the demonstration, there are a number of steps you should take. First of all, you must identify the needs of the people actually going to use the computers. A set of simple questions often leads to a wealth of information which can be further clarified through discussion. It is a good idea to present the questions to the users in writing, which will allow for the organization of thoughts on the subject. For instance, you can start by asking for a description of the needs that the computer is expected to fulfill. The information gained from answering that question will start you off on a search for appropriate software to meet those needs. Once you have identified the software, check with the software publishers to learn on which computers the software can run since not all software runs on all computers. Software compatibility is especially important to consider when looking at computers for administrative purposes. In this area, operating systems (such as CP/M and MS/DOS) frequently vary their input/output system to match specific brands of computers.

The key thing is to eliminate from consideration those computers which cannot use the software you need. Thus, you will need to see far fewer systems demonstrated.

Write the Specifications

At the conclusion of this process, you will have developed a set of

specifications tailored to your requirements. Any computer you see demonstrated will have to meet these specifications. It is a good idea to turn these specifications into a weighted checklist in order to systematically evaluate each system.

Read Industry Reviews

You should read reviews of all computer systems under consideration. Just about every computer has a review published in one source or another. Good reviews which are not highly technical but which zero in on concerns of educators in plain English can be found in such publications as Electronic Learning, Popular Computing, Infoworld, Personal Computing, the EPIE-Consumers Union Pro/Files, as well as in several other publications.

Being familiar with computers from reading reviews will enable you to understand and retain information presented during the demonstration, as well as to raise pertinent questions. In essence, your increase in knowledge will put you more in control of the situation. Also, reading reviews will help to eliminate some computers from being considered for demonstration thus easing your task.

Before the Demonstration

Be sure to alert the vendor to the specific items you want to see demonstrated. Also, it will help if you let the vendor know your level of computer expertise. There is no bigger waste of time than to have a demonstration pitched either too high or too low for your purposes.

AT THE DEMONSTRATION

Be prepared to take an aggressive stand. Don't let the vendor give you a canned sales pitch. Refer to your list of specifications and make certain you see each feature you need demonstrated. Frequently, vendors will have a smooth sales pitch which will lead you down the primrose path with you seeing only what the vendor wants you to see. Don't be a quiet, compliant audience! Make the vendor respond to your concerns to your satisfaction.

Remember, don't accept promises. If you can't see a feature demonstrated, it doesn't exist.

You should get the answers to the following questions before leaving the demonstration and don't be shy about it.

1. Will the system be able to use the software that you have identified?

If at all possible, bring along copies of any software that you will be using. Have the demonstrators boot it up and check to see that it runs properly. This is particularly important if you are considering a look alike machine instead of the computer it is cloned from.

2. How long have the companies you are dealing with been in business?

While many companies in the computer industry are young, you may feel more secure with a company that has been in operation more than a few weeks. Also, before the demonstration, you might want to check such periodicals as The Wall Street Journal, InfoWorld, Computer World, Fortune, and so on, for clues to the financial stability of the computer manufacturer itself. Unfortunately, local computer stores seem to come and go quickly and without notice.

3. How many installations of the computer system being demonstrated have taken place overall and how many in school systems specifically?

Remember, being a pioneer can be a risky business. Make certain you get the names of contact people who have bought the product and see if they are satisfied with it. Don't rely on the

fact that the computer system may have a famous brand name.

4. Is the computer system well made?

Of course, it will be difficult to tear the equipment down on the spot, but you can get a general "feel" for the product. For instance, note if the computer is finished nicely and that all the parts mesh correctly. Make certain that the length of the cables supplied is sufficient to allow you to place equipment where needed. Add a number of peripherals to the computer to see if there is a heating problem caused by an inadequate power supply.

5. Is the computer easy to use?

Check to see if you find your fingers searching for keys that have been relocated to satisfy the whims of the designers. Also, take note of how long it takes to save files and make backup copies. In addition, make certain that you are not suffering eye-strain because of inadequate monitor resolution.

6. How expandable is the computer?

Note whether the computer will accept a variety of peripherals and also that several manufacturers are producing products which will expand the computer's functions. Also, be careful to notice how many of the computer's expansion slots are actually needed for basic computer functions such as the monitor. Slots used in this manner are not really slots available for expansion.

7. Are good quality manuals supplied which are easy to understand?

Remember to carefully inspect the manuals during the demonstration if you haven't done so before. You should be able to find the information you need quickly through the use of a well-sequenced index and table of contents. Also, remember to ask if the manuals are included in the purchase price. They may in fact be a hidden cost (see #9).

8. What will the system cost?

Don't be shy about asking for the educational discount. Most companies dealing with schools provide discounts that go as high as 20 to 25%.

9. What are the hidden costs of the computer system being demonstrated?

While the computer itself may sell for only $200, the price of the necessary peripherals may drive the price several hundred dollars higher.

Or, perhaps the users would like to place several computers in one room in a lab setting. If so, you will need to check how many outlets each computer system under consideration will need and how much amperage is required for each unit. Keep in mind that rewiring rooms to achieve sufficient electrical power can cost several thousand dollars.

10. What kind of warranty is offered?

Despite the fact that products come with "standard" warranties, it is possible to negotiate for extended warranties.

11. What kind of appearance do the people demonstrating the machine make?

They should be knowledgeable, self-assured and dress in a professional manner. Somehow, a sales representative who knows enough not to chew gum during a demonstration and has the sense not to make sexist comments (which seem to be replacing racist comments) has at least the beginnings of a good supportive representative.

Remember, (1) identify the needs the computer is expected to fill, (2) turn those needs into specifications, and (3) make certain that you see the computer demonstrate that it can meet those specifications.

Software Evaluation: Support Through the Texas Education Computer Cooperative (TECC) Project

Vicki S. Smith
Coordinator for Computer-Based Instruction
Region IV Education Service Center
P.O. Box 863
Houston, Texas 77001

The following describes a project funded through the Texas Education Agency for the purpose of increasing the effective use of microcomputers, specifically through courseware evaluation.

The emergence of microcomputer technology in the 1980's has demonstrated its ability to impact the effectiveness of both instruction and learning. The interactive nature of the microcomputer has helped it to gain wide acceptance among educators. As the cost of the hardware has steadily decreased, schools have even more readily accepted the use of such a tool, however, often before proven courseware resources are available to support its use.

As the use of microcomputers expands in educational settings, there is evidence to support the fact that they assist the instructional process effectively only to the extent that quality courseware is available and accessed. There then becomes a great need for careful courseware evaluation, to determine what exactly is "good" and what is inadequate. The quality of courseware is improving, competition is becoming more intense, and greater amounts of money are being spent. The decisions to be made are growing more difficult, and schools must be able to justify such decisions.

Certainly a great responsibility will lie with the trained, computer literate teacher in how the hardware and software is used, but much of the success - or failure - of microcomputer use in the schools will depend upon the courseware selected.

The schools need a set of standards or guidelines whereby they might begin to determine what quality software is. Such guidelines not only assist them in purchasing, but also direct developers and evaluators. Schools need a coordinated effort to provide them such guidelines.

This need is being addressed by a project endorsed by the Texas Education Computer Cooperative. TECC, a group commissioned by the State Commissioner of Education, coordinates computer services in the multiregional and regional processing centers of the state. The group recommended that funds be granted for the creation of a developmental project to address issues related to microcomputer courseware. Acting on the recommendation of the TECC, the Texas Education Agency funded the TABS Related Courseware for Microcomputers Project. The project intent was to increase the effective use of microcomputers, especially as they assist instruction relating to the Texas Assessment of Basic Skills (TABS), the State Board of Education adopted objectives, for reading and/or mathematics

for grades 3, 5, and 9.

The Project Advisory Committee was comprised of one representative from each of the twenty Education Service Centers in Texas. Evaluative criteria and procedures were identified. Region IV Education Service Center, Houston, coordinated training for the twenty representatives, who in turn conducted training in evaluation techniques to hundreds of other teachers across the State. These teachers were then responsible for the evaluation of each courseware package submitted.

Courseware designed for the Apple, Commodore PET and Radio Shack TRS-80 microcomputers was evaluated, and both commercial and non-proprietary sources were included. Region IV Education Service Center invited vendors to participate in the project by providing courseware packages for evaluation, then collected, catalogued, and disseminated courseware and evaluation guides to each of the twenty Education Service Centers.

These evaluation guides were developed following investigation of similar efforts and instruments as MICROSIFT, CONDUIT, National Council of Teachers of Mathematics (NCTM), and EPIE. Courseware evaluations produced data in four general areas: identification, instructional strategies, relations to the Texas Assessment of Basic Skills (TABS), and specific instructional/technical qualities. Each courseware package was evaluated at two different sites for appropriate hardware, instructional design aspects, and correlation to specified skills.

Project work began in February, 1982, and by its conclusion in August, over two hundred courseware packages were evaluated. Data relating to identification were collected and catalogued by Region IV staff assigned to the project. Data addressing instructional strategies and TABS objectives were generated by the evaluation team acting as an entity. Finally, each member of the team individually rated the package on its specific instructional/technical qualities.

At the completion of the evaluation period, all materials were returned to Region IV, where evaluations were tabulated and final results disseminated. Results were compiled using three databases, one each for Apple, Radio Shack, and Commodore PET. These databases, although commercially developed, used formats specifically adapted for the project. Databases were published in both diskette and printed versions. Copies of both versions were disseminated to each Education Service Center in Texas for use by their respective school districts.

The TECC project in its second year of funding, under the project title of the Statewide Microcomputer Courseware Evaluation Network, expanded its focus to include K-12 Reading/Language Arts, Mathematics, Business Education, Vocational Education, and Science. Evaluation largely paralleled that of the first phase. To insure effective evaluations, courseware packages were evaluated at three sites across the state, and the evaluation form saw the inclusion of a student evaluation section. Three hundred additional titles were evaluated.

The Statewide Microcomputer Courseware Evaluation Network Project is currently in its third year. Titles being evaluated this year will include those curricular areas from previous years but will particularly focus on computer literacy materials. Hardware configurations evaluated for have been expanded to include IBM. To better facilitate updating and maintaining of evaluation data, all records are being converted to Bibliographic Retrieval Services (BRS), a commercial database system. Evaluation data will then be available on-line via BRS or through the Education Service Centers.

Production of the database has, obviously been the primary project goal. However, other rewards or accomplishments have been realized as well. Educators

in Texas are becoming more computer literate as they work with courseware evaluation techniques and databases. Education Service Centers and local school districts are accomplishing much by working together. Vendors and software developers who receive copies of evaluations for their submitted packages are discussing ways they might better and more specifically address teacher needs.

The overall intent of the TECC project is not to recommend specific hardware or software, but rather to serve as a tool in the schools' decision-making process of previewing and procuring the appropriate software for instructional use. It is meant to provide information rather than to advocate specific methodology or technique. Hopefully, the product information will serve as a "consumer's guide" to reference, before they purchase, for determining whether the courseware is effective, whether it fits into the instructional and curricular patterns of the district, and whether it addresses specific objectives. The role of the TECC project is one of support.

As microcomputers proliferate in the schools from almost endless vendors, courseware assumes a more and more strategic position. Through the TECC courseware evaluation project, steps are being taken to ensure that microcomputer courseware will be appropriate to the tasks faced by Texas educators.

The AI in CAI

The Application of Artificial Intelligence in Education

Dr. Kathleen M. Swigger
Computer Sciences Department
North Texas State University

The new and exciting area of artificial intelligence has created a lot of speculation as to what the future may become. More specifically, there is some question about how artificial intelligence may affect our present system of education. This paper should serve as an overview, in the hope of giving the participants some perspectives of the field of Intelligent Tutoring Systems (often known as Intelligent Computer-Assisted Instruction, ICAI), particularly for those who may not be fully aware of recent activities.

Educational applications of computers have been under development since the early 1960s. The predominant applications during this period can be categorized as "frame oriented" CAI. In this type of format, the student is usually given some instructional text and asked a question that requires a brief answer. After the student responds, they are told whether the answer is correct. The response is sometimes used to determine the "path" through the curriculum. Whenever the student makes an error, they are branched to a remedial area. When the author designs this type of material, they must anticipate every wrong response, pre-specifying branches to appropriate remedial material based on ideas about what the underlying misconceptions might be that would cause each wrong response.

In spite of widespread use of frame oriented CAI, many researchers believe that these types of CAI systems are not the best use of computer technology. ICAI, on the other hand, tries to go beyond the limits of frame selection and focuses more on the teaching aspects of instructional computing. ICAI systems are quite different from even the most complicated frame oriented branching programs. They incorporate high-level domain expertise, teaching and learning models.

The main components of an Intelligent CAI (ICAI) system are problem solving expertise, the knowledge that the system tries to impart to the student; the student model, indicating what the student does and does not know; and, most importantly, tutoring strategies, which specify how the system presents material to the student (Barr & Feigenbaum, 1982). Not all of these components are fully developed in every system, but most employ partial elements of the three.

The Expert Module

The "expert" part in an ICAI system is charged with the task of generating problems and evaluating the correctness of the student's solutions. This system knowledge of the expert was originally envisioned as a huge, static database that incorporated all the facts of a particular subject to be taught (that is, 5000 history facts, 1000 addition problems, etc.) This type of system has and can be used with some success (Carbonell and Collins, 1973).

More recently, however, authors have suggested that since most knowledge is nonlinear, the data structures for these knowledge bases should reflect different patterns. Instead of static lists of attributes, the data bases now contain the attributes, procedures about the attributes, rules concerning the attributes, etc. In BIP, for example, prerequisite skills were formed into lattices called curriculum netes that related the skills to be taught to sample programming tasks that exercised each skill (Barr, Beard, and Atkinson, 1976).

Student Model Module

The student modelling module represents the student's understanding of the material to be taught. The purpose of modelling the student is to make hypotheses about his misconceptions so that the tutoring module can alleviate the problems. A well written student model should be able to recognize alternative ways of solving problems.

In frame oriented CAI, a pretest is often used to determine a student model of the knowledge. This type of system, however, relies on the ability to model only the probability that students will give a specific response to a stimulus. In general, AI systems attempt to determine what the student understands either by comparing the student to the expert or by locating the student's procedural bugs.

The Tutoring Module

The tutoring module of ICAI systems must integrate knowledge about the student, teaching methods and the subject area. This is the module that communicates with the student, selecting problems for him to solve, monitoring and criticizing his performance, providing assistance, etc. In short, the tutoring module contains knowledge about how to teach.

This is one of the most important parts of the ICAI system and the knowledge contained in this module usually takes the form of rules or teaching strategies. For example, a tutoring module in a program that discusses diagnostic tasks with medical students, has an explicit representation of teaching knowledge. One use of tutoring to select different dialogues

based on domain logic, student knowledge, etc. An example of one of these rules is the following:

If: the goal currently being discussed is true/false parameter
THEN: Generate a question about the expert rule using "facts" format in the premise part and "actual value" format in the action part (Clancey, 1981).

Current Systems

Current intelligent tutorial systems take the form of computer-based 1) problem-solving monitors, 2) coaches, 3) laboratory instructors and 4) consultants. They exist promarily as experimental vehicles on a variety of domains:

1. Symbolic integration (Kimball, 1982).

2. Electronic troubleshooting (Brown, Burton & deKleer, 1981).

3. Axiomatically-based mathematics (Smith et al, 1975).

4. Interpretation of NMR spectra (Sleeman and Hendley, 1981).

5. Medical diagnosis (Clancey, 1981.

6. Informal gaming environment (Burton & Brown, 1981).

7. Program-plan debugging (Miller, 1982).

8. A Consultancy System for users of an Algebraic manipulation system (Genesereth, 1982).

Conclusion

In general, ICAI programs have started to deal with the problem of knowledge about the art of teaching and how this knowledge can be represented and used by a computer program. It is hoped that one day, teachers may go to the computer and simply type in their subject matter and the teaching rules that can be used to manipulate this knowledge. Such a system might look like the example below---

EXPERT MODULE

```
 45     40     312
-21    -30    -243
---    ---    ----
```

STUDENT MODELS

Correct answer model

45-21 = 24

Incorrect answer model

312-243 = 131

TUTORING MODEL

Rule 1
When subtracting numbers in which the bottom number is larger than the top, check to see if the student is subtracting the big number from the small
THEN tell him what he is doing wrong and generate similar problems.

Rule 2
etc.

Bibliography

Barr, A., Beard, M., and Atkinson, R. C. (1976). The Computer as a tutorial laboratory. Int. J. Man-Machine Studies, 8, 567-596.

Barr, A. & Feigenbaum, E. (1982). The Handbook of Artificial Intelligence. Heuristech Press, Stanford, Calif.

Brown, J. S., Burton, R. R., and deKleer,J. (1982). Knowledge engineering and pedagogical techniques in SOPHIE I, II, and III. In D. Sleeman & J. S. Brown (Eds.) Intelligent Tutoring Systems. Academic Press, London.

Burton, R. R. & Brown, J. S. (1976). A tutoring and student modelling paradigm for gaming environments. SIGCSE Bull., 8, 236-246.

Carbonell, J. R. & Collins, A. (1913). Natural Semantics in artificial intelligence. IJCAI, 3, 344-351.

Clancey, W. J. (1981). NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching. IJCAI, 7, 829-836.

Genesereth, M. R. (1982). The role of plans in intelligent tutoring system. In D. Sleeman & J. S. Brown (Eds.) Intelligent Tutoring Systems. Academic Press, London.

Kimball, R. H. (1982). A self-adapting, self-improving tutor for symbolic integration. In D. Sleeman & J. S. Brown (Eds.) Intelligent Tutoring Systems, Academic Press, London.

Miller, M. (1981). A structured planning and debugging environment for elementary programming. Int. Journal of Man-Machine Studies, 11.

Sleeman, D. H. & Hendley, R. J. (1978). Following student reasoning. University of Leeds. Computer Studies Report, 1.

Smith, D. H., & Davis, R. (1981). Frameworks for cooperation in distributed problem solving. IEE Transactions on Sep., Man., and Cybernetics, 11, 1.

Teaching Beginning Students How To Program

Dr. Kathleen M. Swigger

North Texas State University

Computing as a basic subject is appearing in high schools and even some grade schools. The rise of computer science to the status of a real school subject has produced a demand for instruction at a time when little is known either about the nature of computing skills or about effective methods of teaching them. This paper, therefore, attempts to review the major research about the problems of teaching people how to program. It is divided into two sections that address

1) what is known about programming problems

2) what is known about teaching programming.

I. Programming Problems

One of the major characteristics that distinguish a person as a "Master" teacher is their ability to anticipate students' faulty reasoning or bugs. This is also true of Master teachers of computer science. Therefore, it is extremely important that both Computer Science teachers and researchers become aware of the major difficulties that exist when learning a programming language.

One of the first things that young students must learn is how to solve a problem using a computer. Studies (Kreitsberg & Swanson, 1974) indicate that young programmers have problems in understanding both what a computer program can do for them and its relation to the problem. They also find it hard to organize their algorithms so that they actually carry out the required task.

Teachers often say that writing a computer program is just like giving a person instructions, but it is not. For example, most humans who are given instructions can disambiguate seemingly contradictory commands or sentences. Most computers cannot do this task.

Part of a child's difficulty in planning is caused by the disparity between familiar conventions for specifying a plan to a human being, e.g. a cooking recipe, and a computer program. Miller (1975) found that the fundamental difference between the two was that the specification for human use was often "qualification" rather than "conditional". Programs require

created a type of programmers model for learning PASCAL. The most distinguishing feature of Soloway's model is its analysis of looping structures. According to Soloway, students have great difficulty distinguishing among the three looping constructs. The students' major problems, however, are due to their misconceptions of the READ, COUNTER, and TOTAL COUNT structures.

II. Teaching Strategy

One of the difficulties that face the novice is understanding what is going on inside the computer in response to his program. Mayer (1975) showed that novices who had been taught a programming language using a model were better able to read and write program than those who had learned without the model.

Various works (e.g. Cannara) cite the use of games, anathropmorphic metaphors and other aids to understanding programming as factors in their success in teaching and to increase the amount of peer-teaching that occurs. Another hint is to provide analogies, drawn from the experience of the novices, that illustrate some of the novel concepts. This is why many teachers rely on graphics programs to illustrate some of the more traditional programming constructs.

There continues to exist a controversy among computer scientists about the value of using flowcharts. Despite their popularity in teaching texts, Mayer (1975) found that flowcharts only helped novices learning a simplified FORTRAN. Shneiderman (1976) found no difference in the performance of students who had learned programming with or without flowcharts.

Although novices very often write and debug their own programs, Lemos (1978) found that students who debugged their COBOL assignments in groups became more favourably disposed towards programming. Cheney (1977) has also shown that students who worked in pairs on their programming assignments did much better in programming examinations than those who worked individually.

Bork (1971) argues that novices should be taught programming using a "whole program" approach rather than a "grammatical" approach. The whole program approach exposes students to complete programs from the start while the grammatical approach teaches about the constructs in a bottom-up fashion. The validity of this argument was tested by Lemos (1975) but he found no significant differences between groups using each strategy. A synthesis of these two strategies is presented by Shneiderman (1977b) who argues that each on its own is inadequate. His "spiral approach" attempts to provide both the syntactic and semantic understanding of programming, though he presents no empirical evidence of the

students to formulate plans using constructs such as "if .. is true, then ..."

Once a problem has been defined and a plan devised, the student must then code. Many of the recent studies in programming have focused on particular bugs or errors that novices make while programming using specific languages. Analysis of syntactic and semantic errors provides a useful way of categorizing the structure of a language. The studies involving LOGO, BASIC and PASCAL should prove most useful to this particular gathering.

A. LOGO

The most detailed account of children's difficulties is that of Cannara (1976). Some of the students difficulties can be traced back to their misunderstanding of the underlying notional machine. Cannara summarizes their problem as follows:

The two most common, virtually universal misunderstandings of all the students were: (1) misunderstandings of linguistic/ computational context, and (2) ill-defined intents (p. 109).

There were also a number of difficulties associated with the name/value distinction. Difficulties with control included misunderstandings about the recursion/iteration distinction, binding of argument values, result-passing and command-parsing. Syntanctic difficulties included use of predicates, use of value, use of infix operations and invention of illegal "noise" words.

Cannera also found that students were not very good at problem solving, they did not break down problems, nor did they match the problem decomposition to the available language constructs.

B. BASIC

BASIC is perhaps the most widely used programming language in high schools because it is both available and easy to learn. Only recently, however, have we started to look at the errors and bugs students make while learning this language. P. Bayman and R. Mayer (1983) discuss user misconceptions regarding BASIC and programming structures. According to the authors, students have difficulty distinguishing between INPUT and READ statements, between conditional and nonconditional statements, and between equality and assignment statements.

C. PASCAL

PASCAL was recently accepted as the standard CLEP language and is, therefore, receiving wider attention among high school computer science teachers. The most extensive studies of this language have been done by Soloway et al. (1983), who have

success of this method.

Fisher et al. (1973) argue that certain classes of novice e.g. arts students, lack the problem-solving skills and cognitive style to appreciate programming as it is taught today. In consequence, they suggest that an initial programming course should be made especially for these students.

Conclusions

In general, during the planning stage novices have difficulty in formulating an algorithm because they are unfamiliar with the level of generality expected in a computer program and they model their algorithms on more familiar sets of instructions. This later problem leads to a qualificational rather than to a conditional formulation of the task. The teacher can either devote more effort to qualificational/ conditional distinctions or select languages where control mechanisms are closer to human usage.

Novices make numerous errors in the coding stage of programming. These errors are centered around I/O, assignment, looping structures and recursion. Although indentation, blocking and repeating the predicate helps, no one solution appears to alleviate all these problems. One solution might be to use comment lines to reinforce each construct as it is being used. Gradually, the student will become aware of the distinctions among the constructs.

Studies of teaching programming point to the value of providing novices with some sort of model of the machine prior to instruction. Investigations of the role of cognitive style in learning programming indicate the conventional methods of teaching language favors those who are "operation" learners. Also getting students to work in small groups can prove beneficial.

The proposals made here are intended to provide teaching aids that may help us better understand the programming process. Although it is not certain that this represents the mental activities involved, the above research has established the relevance of such work to teachers of computer science. Further work should be available to guide the design of future courses and help us resolve the problems of individuals.

Bibliography

Bayman, P. & Mayer, R. (1983). A diagnosis of beginning programmers' misconceptions of BASIC programming statements. Communications of the ACM, 9, 677-679.

Bark, A. M. (1971), Learning to program for the science student. J. Educ. Data Systems, 8, 1-5.

Connors, A. B. (1976), Experiments in teaching children computer programming. Technical Report No. 271, Institute for Mathematical Studies in the Social Sciences, Stanford University.

Cheney, P. H. (1977), Teaching computer programming in an environment where collaboration is required. J. Assoc. Educ. Data Systems, 11, 1-5.

Fisher, P. Hankley, W. and Wallentine, V. (1973). Separation of introductory programming and language instruction. SIGCSE Bull., 5, 9-14.

Krietsberg, C. B. & Swanson, L. (1974). A cognitive model for structuring an introductory programming curriculum. Proceedings of the AFLPS National Computer Conference, 43, 307-311.

Lemos, R. S. (1975). FORTRAN Programming: an analysis of pedagogical alternatives. J. Educ. Data Processing, 12, 21-29.

Mayer, R. E. (1975). Different problem-solving competencies established in learning computer programming with and without meaningful models. J. Educ. Psychol., 67, 725-734.

Miller, L. A. (1974). Programming by Non-programmers. Int. J. Man-Machine Stud., 6, 237-260.

Schneiderman, B. (1976). Exploratory experiments in programmer behavior. Int. J. Computer Information Sci., 5, 123-143.

Administrative Applications of Microcomputer Software for Physical Education, Sport and Athletics

by
Janice C. Wendt and James R. Morrow, Jr.
University of Houston, Houston, Texas, 77004

Abstract

Software is available to record sport performance, evaluate physical parameters, health data, nutrition, scouting statistics, sport analysis, terminology, offensive and defensive strategies, and tournament scheduling. Before purchasing software it is important to determine if it is user friendly, has appropriate documentation and is flexible enough to meet user demands.

Microcomputer software is available for sport, health and well-being. If one can turn on a machine, insert a disk, and can at least function with a "hunt-and-peck" typing system, one can effectively use a microcomputer. There is no need to feel inadequate in working with microcomputers since most of the available software is "user friendly" which enables the administrator, teacher, coach or student to effectively use the programs. The purpose of this presentation is to make coaches, administrators and teachers aware of the many valuable uses of the microcomputer in administrative functioning and sport performance. Additional purposes are to discuss the availability of existing software to address specific needs of sport performance and to provide the new user with criteria with which to evaluate software.

Software currently available can be used to record sport performance, evaluate physical parameters (e.g., body composition,

percent body fat, Max VO_2), monitor training programs, and model biomechanical analysis of performance. Fitness tests, health data, nutrition, weight gain or reduction, and personalized exercise programs are other examples of available software. The coach can also take advantage of software developed for scouting purposes in basketball, baseball, football, etc. The reduction of time spent in paperwork and the accuracy of the data make microcomputer software a cost-efficient item and frees the coach, teacher or administrator for quality time with the individuals involved.

The Texas Test of Motor Ability and Physical Fitness, Youth Fitness Test, and Health Related Fitness Test have all been put into software format and are "user friendly" so that even the students, elementary or secondary, can input their scores. The computer can be wheeled into the gymnasium and as tests are completed, students can input their scores and obtain a written evaluation of their performance in a matter of seconds. This saves the teacher class time as well as record keeping time which also risks errors not only in the translation but in comparing the scores with the national, state, or local norms.

If body composition measurements are desired, skinfold measurements can be taken and entered into the computer for an estimate of body fat and desired weight at different percent levels. If an underwater weighing tank is available, the body surface area, body volume, body fat and density, fat and lean body weight and desirable body weight can be obtained by microcomputer.

Physicians and exercise physiologists also use microcomputers. A more sophisticated system of analysis can be

achieved by interfacing a microcomputer and printer with an electrocardiograph, analog to digital board, motor-driven treadmill and a custom pulse conditioner (Donnelly, 1983). This provides the tester with the capability of choosing any protocol for exercise with each exercise parameter automatically regulated. Blood pressure is entered and a print-out of one's graded exercise record with predicted Max VO_2, heart rate, energy expenditure, intensity of exercise and ECG data are all present. The microcomputer is currently being used in many Wellness clinics and exercise centers. Less sophisticated methods involve assessing physiological parameters and then entering the values into the computer in order to calculate summary data.

Nutrition software provides a variety of capabilities (Halligan, 1984). Comparing a nutritive analysis of single foods, combinations of food, meals, daily menus, and complete diets with the RDA for age, weight, height, and gender, a print-out graphically displays the results (THE NUTRITIONIST). Other programs permit one to analyze food intake, calculate the ideal caloric intake based on age, gender, activity level, and desired weight goals (NUTRICALC), and then calculates what the weekly caloric intake should be to reach or maintain the desired weight. Individualized exercise prescriptions are also a part of the microcomputer format. INSHAPE is a program with a combination of exercise and diet using Dr. Kenneth Cooper's aerobic point system giving one a display of weight versus caloric intake or exercise for 60 days or 52 weeks.

For training purposes several software programs have potential for the athlete. Analyzing performance by several

different criteria such as temperature, humidity, pulse, time, and motivation, THE RUNNING LOG provides bar graphs showing daily mileages, a diary of running days, and monthly graphs of pulse in relation to daily mileages. A more sophisticated technique used at the Olympic Training Center in Colorado Springs, Colorado proposes to actually maximize sport performance and analyze what can be done to improve the movement (Friefeld, 1983). First high-speed film is taken of the athlete and it is digitized (projected on a tablet somewhat akin to electronic graph paper over which one would locate a cursor on a particular point and press a button). This data can then be analyzed with various programs which result in recommendations to the athlete for efficient movement and performance improvement.

Sports strategies are programmed into software quite efficiently (Barlow & Bayalis, 1983). Volleyball and racquetball offensive and defensive strategies are used to enhance performance for beginning players. Terminology for individual/dual/team sports and aquatic sports can be learned via microcomputer. Scoring, skill fundamentals and spare conversions can be learned in bowling, and skill analysis and safety procedures can be learned in swimming rescue, winter sports, aquatic sports as well as individual/dual/team sports.

The athletic coach will enjoy the use of a tournament scheduling package which accomodates round robin, single or double elimination with seeding options, heat, lane, and flight assignments. Additionally, basketball, baseball and football scouting statistics have been actualized on the microcomputer freeing the coach or manager much time spent in calculating

percentages and probabilities. By scouting field goals attempted and made, offensive rebounds, defensive rebounds, steals, violation, bad passes, turnovers, etc. a hard copy can be obtained for each player. The same is true in football for elaborate analysis of offensive and defensive success, first down, fourth down patterns, etc. which provide the coach a unique profile not as easily obtained by hand calculations. Output from such programs can impact on field decisions based upon the quick response available via the microprocessor.

The new user must have criteria with which to judge the merits of one software package over another. Questions worth asking include: How easy is it to input the data? Will the data provided by the program be meaningful? How extensive is the documentation included with the program? How flexible is the program?

Documentation or tutorials may be provided with the software. If one has trouble reading or understanding these instructions, it is also probably correct to assume that one would also have similar problems using the software itself. The most important criterion of all is that the software is "user friendly." This microcomputer term implies the inclusion of appropriate positive feedback, directions, and hints or help menus in case the user is unsure of what to do. After the decision has been made about the kind of software needed, one should check catalogs, magazines, commercial vendors, regional media centers and local colleges or universities to locate the programs which best meet one's desires.

Microcomputers can be an invaluable asset to any organization and no longer need to be rejected due to lack of knowledge or

"microcomputer phobia". They can provide quick and accurate results in a most efficient manner.

REFERENCES

Barlow, D. A., & Bayalis, P. A. Computer facilitated learning. Journal of Physical Education, Recreation and Dance, 54:9 (December 1983), 27-29.

Donnelly, J. E. Physiology and fitness testing. Journal of Physical Education, Recreation and Dance. 54:9 (December 1983), 24-26.

Freifeld, K. Body management. Personal Computing. (August 1983), 60-71+.

Halligan, J. M. Software to keep you physically fit. Software Supermarket. (January 1984), 18-25.

IS THERE A PLACE FOR VISICALC IN THE ELEMENTARY SCHOOL CLASSROOM?

Jean W. Wilson, Ed.D.
Professor of Education
Trinity University
715 Stadium Drive
San Antonio, Texas 78284

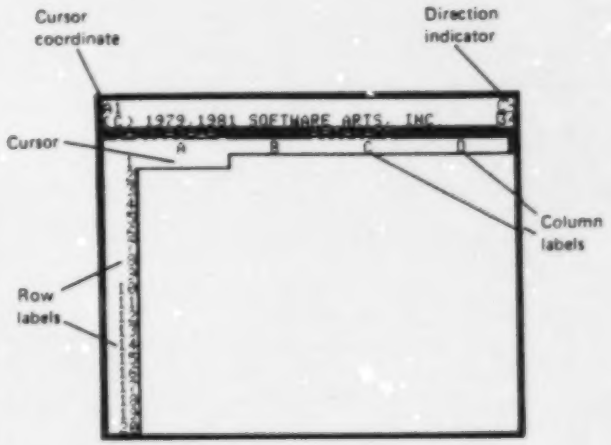
Some people think that VisiCalc has no place in the elementary school classroom; however, children in the intermediate grades can gain better insight into some mathematical concepts as a result of working with VisiCalc.

VisiCalc is a tool that has achieved widespread use and extreme popularity in recent years. It is an electronic spreadsheet that is used extensively in business. In this session participants will be shown what VisiCalc is, how to use it, and some applications for the elementary classroom. Classroom applications will be in the areas of mathematics, science, and social studies.

VisiCalc is a versatile software tool that has achieved widespread use and extreme popularity in recent years. It is an electronic spreadsheet that is used extensively in business. Its power lies in its ability to provide the capabilities normally found in the use of a pencil, a sheet of paper, and a calculator to give results accurately, quickly, and with great flexibility. VisiCalc is widely used in business for budgeting and forcasting.

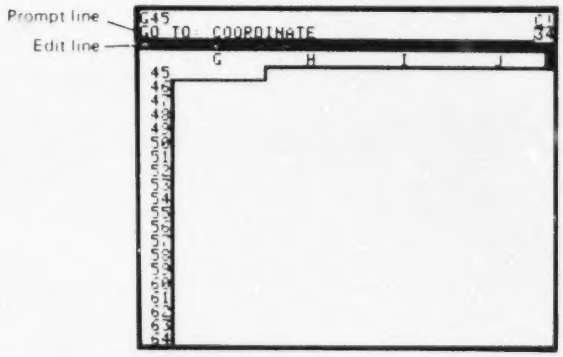
We are not considering whether or not VisiCalc should be taught to elementary school students, but whether or not it could be taught to elementary school students. There are applications that could be used in the upper elementary grades that would help students better understand mathematical relationships and the use of formulas, and aid in problem solving and decision making. In a problem solving situation the student is able to see how changing one entry can change many outcomes. VisiCalc can help answer many "what if..." questions by letting the student change an entry and then quickly showing the results. The use of VisiCalc would provide students an experience with software that is used as a tool and which probably is different from software they have been using.

The writer assumes that the reader is able to load VisiCalc into the memory of the computer. Once this is done the screen will look like the one in Figure 1.



You see just one small part, a window, of a large worksheet. You can see columns A, B, C, D and rows 1, 2, 3, ..., 20 on the screen. (Actually, there are 63 columns and 254 rows on the worksheet.) These labels permit identification of locations at the intersection of columns and rows. For example, the intersection of column A and row 1 is labeled A1. The rectangle that is in the A1 position in this figure is called the cursor and serves as a highlight of one entry. The cursor must be at the location where the data is to be entered. To move the cursor to the right, press the right arrow key. The cursor should now be positioned at B1. To move the cursor back to A1, press the left arrow key. Look at the top left corner of the screen to see the name, or coordinate, of the cursor position. If the computer you are using does not have four separate arrow keys, press the space bar to change the cursor movement from left and right to up and down. To go back to left and right movement, just press the space bar again. The direction of the cursor movement can be seen in the top right corner of the screen.

VisiCalc provides a quicker procedure to move the cursor from one entry to another. Press the key, after which the screen looks like Figure 2.



The second line at the top of the screen is a prompt line which indicates what is to be done next. The third line on the screen is the edit line which shows what is being entered. After pressing the > key, type in the coordinate of the desired location. Then press RETURN. The cursor will then be at the desired location.

There are two ways to erase an error that has been made in an entry. To erase an error before the RETURN key is pressed, press the ESC key once to erase each character. If the RETURN key has already been pressed, the entry can be erased by pressing /B. In some cases VisiCalc will detect an error as it is entered, issue a double beep, and cancel the entire command.

After the student learns the mechanics of using VisiCalc, there are several skills that can be developed. One beneficial skill is setting up the formulas to find totals, subtotals, differences, percents, etc. Many mathematical skills can be reinforced with these activities. Another skill is predicting, before changing an entry, what would happen if the entry is changed. Deciding which entry is to be changed and now that entry is to be changed provides practice with decision making.

There are many problems related to math, science, or social studies which can be used for developing these skills. Included in this paper are several examples.

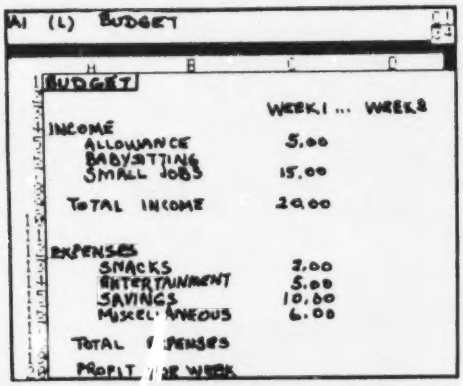
This first example of working with a budget will be used to help the child learn the basic skills of working with VisiCalc.

1. Have the students boot the VisiCalc diskette.

2. Remove the VisiCalc diskette from the disk drive, and insert the storage diskette.

3. To load the program desired, type /SL and then the name of the program: in this case BUDGET. Press the RETURN key. When the program is loaded the screen should look like

Figure 3.



4. Give the student practice in moving the cursor to specific coordinates: Move the cursor to A12. What information is in that coordinate? (Refer to the preceding paragraphs to see how to do this.)

5. Give the student practice in entering data: Enter 12 at C6. Notice what this does to the entry in C9.

6. Give the student practice in changing the data in an entry: Move the cursor to C7 and type in 15. Notice that 15 replaces what was previously in the coordinate. Notice how this changes C9.

7. Give the student practice in writing formulas: Move the cursor to C9. Look at the top line of the screen. Remember C9 indicates the coordinate where the cursor is located. (V) indicates that a value (rather than a label) is typed in. The next information on the top line is the formula: +C5+C6+C7. The formula is the sum of the values in each coordinate. The "+" must be typed in front of the letter C to tell VisiCalc that this is a value rather than a label. Have the students write the formula in C18 that would give the total of the expenses. Be sure the cursor is on C18 when the formula is being written. Move the cursor to the coordinate that would indicate the profit for the week. Let the student decide what the formula would be and type it in.

This problem could be expanded by increasing the weeks' income by 10%. An additional column would be set up to show the 10% increase. If the student is just concerned about a 10% increase of his total income, a formula, C9*1.10, can be written in the D9 coordinate. If a 10% increase for each source of income is being considered, the formula, C5*1.10, can be written in D5 and then the replication command can be utilized to use the same formula for all income entries. The replication formula is less obvious and should be used after the student is fairly proficient with writing formulas. To utilize the replication command in this example type /R:D6.D9 and press RETURN. Then type R. Have the student notice the formula in each entry: D5, D6, D7, D8, D9. Remember the formula can be seen at the very top of the screen. Have students decide how expenses could be decreased by 20%?

To use VisiCalc to solve problems it is necessary to create models or patterns which are called templates. A template was already written in the previous example.

A problem that could be used to give the student experience with making a template and writing formulas would be one like this problem of a dripping faucet. If a faucet drips 10 drops per minute and there are 20 drops per milliliter, what is the capacity of water wasted in one hour, one day, one week, one month? The labels that could be typed in are:

			WATER (IN LITERS) WASTED:			
Drops	mL	L	Hour	Day	Week	Month

The student could first do the computation to find the answer when the drip

is 10 drops per minute. Then the student should generalize the problem by writing formulas that could be used for any number of drips. For example, to write the formula to find mL, move the cursor to B7 and type in the formula A7/20. The number .5 will appear in the coordinate B7. If 30 is in A8, the formula for B8 would be A8/20 and 1.5 would be in the coordinate. The replication formula could be used to find mL for any number of drops given. If there are twelve entries in column A, place the cursor on B7 and type /R:B8. B18. Press RETURN and R. As the student determines formulas to obtain the other needed information, he/she continues to reinforce math concepts that have been learned.

Record keeping is always very time consuming in an individualized program . VisiCalc could be used to keep a record of grades and percent of responses that are correct. A template could be made in which individual assignments are listed, with a column for the number of correct responses, a column for the number of possible responses, and another column for the percent of correct responses. Students could have the benefit of keeping up with their progress in the specific program as they work with VisiCalc.

There are many problems in science that lend themselves to giving the student experience in making predictions. For example, if a candle is burned in a one liter contained, how long will the candle burn? What would happen if the size of the container is doubled, tripled? The experiment would need to be performed to confirm the prediction, and then the student could generalize a formula for any size container and with VisiCalc get all the answers almost immediately.

An example in social studies for use of VisiCalc would be in a unit on the Westward Movement. If the settlers had a given amount of money to make the trip west, a budget could be made for food, clothing, ammunition, supplies, miscellaneous. Hypothetical actual expenses could be compared with anticipated expenses, and predictions could be made as to how long the money would last. Many "what if ..." questions could be considered in this situation.

The use of VisiCalc in the elementary school can be started on a very simple level as described in this paper. After the student becomes proficient with the basic skills, he/she should be challenged to find additional uses of VisiCalc to help obtain solutions to problems that may be encountered. There are many possibilities. There is also much more potential for VisiCalc in the elementary school. For example, VisiCalc has built-in functions which give VisiCalc more power and can provide the student with a better understanding of functions. An excellent book for the student of VisiCalc is THE VISICALC BOOK by Donald H. Bell. This book is written in several editions to be compatible with the computer that is being used.

Reference

Bell, Donald H. THE VISICALC BOOK, APPLE EDITION. Reston, VA: Reston Publishing Company, Inc., 1982.